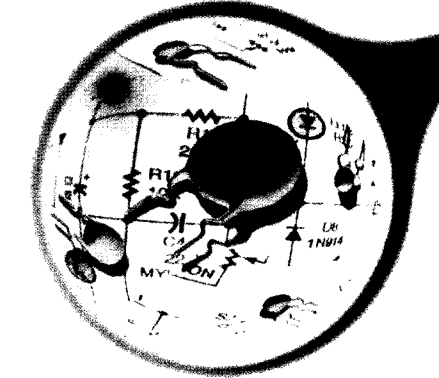
Наиболее полное справочное руководство

Михаил Гук

Стандарты и расширения
Программирование
Конструктивные элементы
Схемотехника



Интерфейсы ПК
Справочник

ПИТЕР

Интерфейсы ПК
Справочник

Это наиболее полный справочник, содержащий всю необходимую информацию, которая может понадобиться разработчику периферийных устройств. Вам не придется перерывать горы литературы или тратить свои деньги на поиск информации в Интернете для того, чтобы найти ответ на интересующий вас вопрос.

Удобное расположение и систематизация материала и подробный алфавитный указатель делают книгу незаменимым консультантом, который всегда под рукой на вашем рабочем столе.

«Интерфейсы ПК: справочник» – настольная книга и профессионала-разработчика, и программиста, и опытного пользователя!

Уровень пользователя

· Начинающий Опытный Профессионал

Категория книги: Справочник
Аппаратные средства

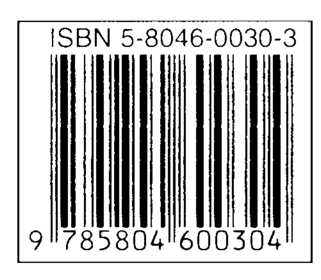
Информация, которую вы найдете в справочнике:

- **Параллельные интерфейсы (Centronics и LPT-порт)**
- **Стандарт IEEE 1284-1994**
- **Последовательные интерфейсы (RS-232C, MIDI, «токовая петля»)**
- **Специальные интерфейсы (клавиатура, мышь, GAME-порт)**
- **Аудиоинтерфейсы**
- **Интерфейсы мониторов (RGB TTL, analog, EVC)**
- **Шины расширения (ISA, EISA, PC-104, MCA, VLB, PCI, AGP, PCMCIA)**
- **Интерфейс ATA, ATAPI, SCSI**
- **Последовательные шины (USB, FireWire, ACCESS, Bus, I²C)**
- **Системотехника IBM PC (память, ввод/вывод, прерывания, Plug and Play)**
- **Конструктивные элементы компьютеров и интерфейсов**
- **Элементы цифровой схемотехники**

и многое другое

Лучший подарок

Михаил Гук



ISBN 5-8046-0030-3
9 785804 600304

На каждый вопрос — готов ответ!



Михаил Гук

Интерфейсы ПК

Справочник

Санкт-Петербург
Москва • Харьков • Минск
1999



Михаил Гук

Интерфейсы ПК: справочник

Главный редактор	*В. Усманов*
Заведующий редакцией	*Е. Строганова*
Ведущий редактор	*А. Пасечник*
Литературный редактор	*Ф. Андреев*
Художественный редактор	*И. Половодов*
Иллюстрации	*О. Гук*
Корректоры	*Н. Рощина, С. Шевякова*
Верстка	*П. Быстров*

ББК 32.973.2-04я22

УДК 681.326(03)

Гук М.

Г93 Интерфейсы ПК: справочник — СПб: ЗАО «Издательство «Питер», 1999. — 416 с.: ил.

ISBN 5-8046-0030-3

Наиболее полный и подробный справочник по всем внутренним и внешним интерфейсам персонального компьютера: последовательного и параллельного портов, инфракрасного порта, IDE, SCSI, шинам ISA, EISA, MCA, VESA, PCI, AGP, PC-Card (PCIMCIA) и др. В книге приводятся полные описания и спецификации интерфейсов: от адресов ввода/вывода и прерываний до уровней сигналов и контактов разъемов.

Для программистов, инженеров и разработчиков периферийного оборудования.

ISBN 5-8046-0030-3

ЛР № 066333 от 23.02.99 г.

Подписано к печати 23.02.99. Формат 84x108 $^{1}/_{32}$. Усл. п. л. 43,68.
Доп. тираж 7000 экз. от 03.03.99 г. Заказ № 530.

ЗАО «Издательство **«Питер»**. 196105, Санкт-Петербург, Благодатная ул., 67.

Отпечатано с готовых диапозитивов в ГИПК «Лениздат»
(типография им. Володарского) Государственного комитета РФ по печати.
191023, С.-Петербург, наб. р. Фонтанки, 59.

Содержание

Предисловие

Совсем недавно была закончена работа над книгой «Аппаратные средства IBM PC. Энциклопедия». То, что Жванецкий говорил по поводу ремонта, справедливо, пожалуй, и для любой книги по компьютерной тематике — работу над ней «нельзя завершить, а можно только прекратить», поскольку ее написание «не процесс, а состояние». Вскоре после того, как автора перестали подпускать к тексту «Энциклопедии» (иначе книга не вышла бы в свет), ему удалось раздобыть новый материал по интерфейсам, что послужило толчком для написания новой книги. Это уже вторая «дочерняя» книга, первая — «Процессоры Intel: от 8086 до Pentium II» на полгода опередила свою родительницу и имела успех у читателей. Здесь тоже чего-то больше, что-то дано глубже, чем в «Энциклопедии», но круг вопросов ограничен только аппаратными интерфейсами утилитарного назначения. Книга адресована широкому кругу читателей, так или иначе связанных в своей деятельности с компьютерами. Знание различных интерфейсов позволяет разработчику аппаратуры выбирать вариант, адекватный решаемой задаче. Тому, кто занят не разработкой устройств, а их интеграцией, знание некоторых подробностей об интерфейсах тоже поможет в выборе изделий. Они, например, получат ответ на вопрос, почему и когда за SCSI нужно платить больше денег, чем за ATA/ATAPI. Сведения, приведенные в этой книге, пригодятся и программистам-системщикам, которым приходится разрабатывать собственные драйверы периферийных устройств или залезать в чужие творения. Любознательные читатели смогут из книги почерпнуть много интересных подробностей «из жизни интерфейсов», например, почему так хорошо (или плохо, нужное подчеркнуть) работает система PnP для шины ISA.

Для удобства восприятия в книге принята система текстовых выделений. *Курсивом* выделены ключевые слова (например, первый раз встречающиеся определения), а также на-

звания *состояний*, в которых могут пребывать некоторые объекты. Названия электрических сигналов, например IRQ2, можно отличить от категорий, близких программистам, например инструкций *INT 2*, или имен регистров (*MSR*). Для тех, кому «электрические детали» покажутся не совсем понятными, рекомендую обратиться за помощью к приложениям, в которых приведены некоторые сведения по системотехнике PC.

Нет возможности изложить все нюансы интерфейсов — тогда книга оказалась бы или слишком толстой, или узко специализированной. Интернет дает возможность добраться до многих спецификаций и статей — спасибо коллективу RUSNet (http://www.neva.ru). За время работы над книгой была собрана немалая коллекция документов, открытых для доступа (правда, большинство из них на английском языке). В систематизированном виде (как файлы HTML) эта коллекция (или ссылка на нее) будет помещена на сервере издательства «Питер» http://www.piter-press.ru.

Связаться с автором можно по электронной почте: *mgook@stu.neva.ru*.

Введение

Толковый словарь по вычислительным системам определяет понятие *интерфейс* (*interface*) как границу раздела двух систем, устройств или программ; элементы соединения и вспомогательные схемы управления, используемые для соединения устройств. Эта книга посвящена интерфейсам, позволяющим подключать к персональным (и не только) компьютерам разнообразные периферийные устройства (ПУ) и их контроллеры.

По способу передачи информации интерфейсы подразделяются на параллельные и последовательные. В *параллельном* интерфейсе все биты передаваемого слова (обычно байта) выставляются и передаются по соответствующим параллельно идущим проводам одновременно. В PC традиционно используется параллельный интерфейс *Centronics*, реализуемый LPT-портами. В *последовательном* интерфейсе биты передаются друг за другом, обычно по одной линии. COM-порты PC обеспечивают последовательный интерфейс в соответствии со стандартом *RS–232C*.

При рассмотрении интерфейсов важным параметром является *пропускная способность*. Технический прогресс приводит к неуклонному росту объемов передаваемой информации. Если раньше матричные принтеры, печатающие в символьном режиме, могли обходиться и COM-портом с невысокой пропускной способностью, то современным лазерным принтерам при высоком разрешении не хватает и производительности самых быстрых LPT-портов. То же касается и сканеров. А передача «живого» видео, даже с применением компрессии, требует немыслимой ранее пропускной способности.

Вполне очевидно, что при одинаковых быстродействии приемопередающих цепей и пропускной способности соединительных линий по скорости передачи параллельный интерфейс должен превосходить последовательный. Однако

повышение производительности за счет увеличения тактовой частоты передачи данных упирается в волновые свойства соединительных кабелей. В случае параллельного интерфейса начинают сказываться задержки сигналов при их прохождении по линиям кабеля, и, что самое неприятное, задержки в разных линиях интерфейса могут быть различными вследствие неидентичности проводов и контактов разъемов. Для надежной передачи данных временные диаграммы обмена строятся с учетом возможного разброса времени прохождения сигналов, что является одним из факторов, сдерживающих рост пропускной способности параллельных интерфейсов. В последовательных интерфейсах, конечно же, есть свои проблемы повышения производительности, но, поскольку в них используется меньшее число линий (в пределе — одна), повышение пропускной способности линий связи обходится дешевле. Может, кто-нибудь еще помнит интерфейс канала ЕС ЭВМ (два пучка коаксиальных кабелей толщиной в руку) и видел кабель USB — эти интерфейсы (параллельный и последовательный) имеют примерно одинаковую пропускную способность.

Для интерфейса, соединяющего (физически или логически) два устройства, различают три возможных режима обмена — дуплексный, полудуплексный и симплексный. *Дуплексный* режим позволяет по одному каналу связи одновременно передавать информацию в обоих направлениях. Он может быть асимметричным, если пропускная способность в направлениях «туда» и «обратно» имеет существенно различающиеся значения, или симметричным. *Полудуплексный* режим позволяет передавать информацию «туда» и «обратно» поочередно, при этом интерфейс имеет средства переключения направления канала. *Симплексный* (односторонний) режим предусматривает только одно направление передачи информации (во встречном направлении передаются только вспомогательные сигналы интерфейса).

Другим немаловажным параметром интерфейса является *допустимое удаление* соединяемых устройств. Оно ограничивается как частотными свойствами кабелей, так и помехозащищенностью интерфейсов. Часть помех возникает от соседних линий интерфейса — это перекрестные помехи,

защитой от которых может быть применение витых пар проводов для каждой линии. Другая часть помех вызывается искажением уровней сигналов.

С появлением шин USB и FireWire в качестве характеристики интерфейса стала фигурировать и *топология соединения*. Для интерфейсов *RS-232C* и *Centronics* практически всегда применялась *двухточечная топология* PC — устройство (или PC — PC). Исключениями из этого правила являются различные устройства безопасности и защиты данных (*Security devices*), которые подключаются к COM- или LPT-портам, но имеют разъем для подключения внешнего устройства. Но поскольку эти устройства для традиционной периферии прозрачны, можно считать, что они не нарушают общего правила. Аналогично обстоит дело и с адаптерами локальных сетей (например, Paraport) и внешних дисковых накопителей (Iomega Zip), подключаемых к LPT-портам. Хотя разрабатываемые стандарты для параллельного порта (IEEE 1284.3) и предусматривают соединение устройств в цепочку (*Daisy Chain*) или через мультиплексоры, широкого распространения такие способы подключения пока не получили. К другому классу исключений относится построение моноканала на COM-портах, которое несколько лет назад применялось в «любительских» локальных сетях, но было вытеснено существенно более эффективной и подешевевшей технологией Ethernet. Интерфейсные шины USB и FireWire реализуют *древовидную топологию*, в которой внешние устройства могут быть как оконечными, так и промежуточными (разветвителями). Эта топология позволяет подключать множество устройств к одному порту USB или FireWire.

Важным свойством интерфейса, на которое часто не обращают внимания, является *гальваническая развязка*, а точнее — ее *отсутствие*. «Схемные земли» устройств, соединяемых интерфейсом с COM- или LPT-портом PC, оказываются связанными со схемной землей компьютера (а через интерфейсный кабель и между собой). Если между ними до подключения интерфейса была разность потенциалов, то по общему проводу интерфейса потечет уравнивающий ток, что плохо по целому ряду причин. Падение напряжения на общем про-

воде, вызванное протеканием этого тока, приводит к смещению уровней сигналов, а протекание переменного тока приводит к сложению полезного сигнала с переменной составляющей помехи. К этим помехам особенно чувствительны ТТЛ-интерфейсы; в то же время в *RS-232C* смещение и помеху в пределах 2 В поглотит зона нечувствительности. В случае обрыва общего провода или плохого контакта, а гораздо чаще — при подключении и отключении интерфейсов без выключения питания устройств разность потенциалов прикладывается к сигнальным цепям, а протекание уравнивающих токов через них часто приводит к пиротехническим эффектам. Откуда берется эта разность потенциалов, объяснить нетрудно (см. приложение Г). Из рассматриваемых интерфейсов гальваническую развязку устройств обеспечивают только MIDI (одностороннюю с напряжением изоляции до 100 В) и шина FireWire (полную с напряжением изоляции до 500 В).

Внутренние интерфейсы

Вышеприведенные рассуждения относились в основном к *внешним* интерфейсам — соединяющим отдельные устройства, удаленные друг от друга на заметное расстояние. Но интерес представляют и *внутренние* интерфейсы, предназначенные для быстрой связи на короткие расстояния. Стандартизованные *шины расширения ввода/вывода* обеспечивают расширяемость PC, который никогда не замыкался на выполнении сугубо вычислительных задач. Эти шины предоставляют более широкие возможности для взаимодействия процессора с аппаратурой, не скованные жесткими ограничениями внешних интерфейсов. Шины расширения ввода/вывода реализуются в виде слотов (щелевых разъемов) на системной плате компьютера. К ним относятся:

- ISA-8 и ISA-16 — традиционные универсальные слоты подключения периферийных адаптеров, не требующих высоких скоростей обмена (раньше ISA была единственной шиной расширения).
- EISA — дорогая (по стоимости и системной платы, и плат расширения) 32-битная шина средней производительности, применяемая в основном для подключения контрол-

леров дисков и адаптеров локальных сетей в серверах. В настоящее время вытесняется шиной PCI, хотя и применяется в серверах, где необходимо установить множество плат расширения (системную плату, у которой слотов PCI больше, чем 4, найти довольно трудно, а для шины EISA 6–8 слотов — явление обычное).

- MCA — шина компьютеров PS/2, до сих пор применяемая и в некоторых серверах. Производительность — средняя. Адаптеры для шины MCA не получили широкого распространения.
- VLB — быстродействующая 32- или 64-битная локальная шина процессора, применявшаяся в среднем поколении системных плат для процессора 486. Используется для подключения контроллеров дисков, графических адаптеров и контроллеров локальных сетей в паре со слотом ISA/EISA. С процессорами последующих поколений не применяется.
- PCI — самая распространенная высокопроизводительная 32/64-битная шина, применяемая в компьютерах на процессорах 486 и старше, а также на «неинтеловских» платформах. Используется для подключения адаптеров дисков, контроллеров SCSI, графических, коммуникационных и других адаптеров. На системной плате чаще всего устанавливают 3 или 4 слота PCI.
- AGP — магистральный интерфейс подключения интеллектуальных графических адаптеров, применяемый в новых системных платах для процессоров шестого поколения.
- PC Card, он же PCMCIA — слот расширения блокнотных компьютеров, который в принципе может присутствовать и в компьютерах настольного исполнения (встречать на практике не доводилось). Предназначен для подключения периферии к блокнотным PC.

За универсальность и производительность внутренних шин расширения приходится расплачиваться более замысловатой реализацией интерфейсных схем и сложностями при обеспечении совместимости с другим установленным в компьютер оборудованием. Здесь ошибки могут приводить к потере (хорошо, если временной) работоспособности компьютера.

Недаром серьезные производители компьютеров гарантируют работоспособность своих изделий только при установке сертифицированных (ими или независимыми лабораториями) карт расширения. При использовании внешних интерфейсов неприятности в случае ошибок чаще всего имеют отношение только к подключаемому устройству. Хотя и здесь случаются всякие «чудеса», часть из которых описана в приложении Г.

Своеобразное положение занимает шина SCSI — интерфейсная шина системного уровня, предназначенная для подключения широкого спектра ПУ, требующих высокой скорости обмена данными. Конструктивно эта шина реализуется ленточным кабелем-шлейфом, соединяющим внутренние и внешние устройства с хост-адаптером компьютера. По функциональным возможностям и производительности за этой шиной «гонится» похожая по конструкции шина ATA, которая из специализированного интерфейса дисковых накопителей выросла до вполне универсального интерфейса ATAPI, логически родственного SCSI. Однако, в отличие от SCSI, ATA конструктивно является сугубо внутренней, а по функциональным возможностям (количеству подключаемых устройств, обеспечению многозадачности) шину SCSI ей, похоже, не догнать.

На этом мы закончим краткий обзор интерфейсов и перейдем к их детальному описанию.

1. Параллельные интерфейсы

Параллельные интерфейсы характеризуются тем, что в них для передачи бит в слове используются отдельные сигнальные линии, и биты передаются одновременно. Параллельные интерфейсы используют логические уровни ТТЛ (транзисторно-транзисторной логики), что ограничивает длину кабеля из-за невысокой помехозащищенности ТТЛ-интерфейса. Гальваническая развязка отсутствует. Параллельные интерфейсы используют для подключения принтеров. Передача данных может быть как однонаправленной (*Centronics*), так и двунаправленной (*Bitronics*). Иногда параллельный интерфейс используют для связи между двумя компьютерами — получается сеть, «сделанная на коленке» (LapLink). Ниже будут рассмотрены протоколы интерфейсов *Centronics*, стандарт IEEE 1284, а также реализующие их порты PC.

1.1. Интерфейс Centronics и LPT-порт

Для подключения принтера по интерфейсу *Centronics* в PC был введен *порт параллельного интерфейса* — так возникло название *LPT*-порт (Line PrinTer — построчный принтер). Хотя сейчас через этот порт подключаются не только построчные принтеры, название «LPT» осталось.

1.1.1. Интерфейс Centronics

Понятие *Centronics* относится как к набору сигналов и протоколу взаимодействия, так и к 36-контактному разъему на принтерах. Назначение сигналов приведено в табл. 1.1, а временные диаграммы обмена с принтером — на рис. 1.1.

Интерфейс *Centronics* поддерживается принтерами с параллельным интерфейсом. Его отечественным аналогом является интерфейс *ИРПР-М*.

Традиционный порт *SPP* (Standard Parallel Port) является однонаправленным портом, через который программно реализуется протокол обмена *Centronics*. Порт вырабатывает

аппаратное прерывание по импульсу на входе Ack#. Сигналы порта выводятся на *разъем DB-25S* (розетка), установленный непосредственно на плате адаптера (или системной плате) или соединяемый с ней плоским шлейфом. Название и назначение сигналов разъема порта (табл. 1.2) соответствуют интерфейсу *Centronics.*

Таблица 1.1. Сигналы интерфейса Centronics

Сигнал	I/O*	Контакт	Назначение
Strobe#	I	1	Строб данных. Данные фиксируются по низкому уровню сигнала
Data [0:7]	I	2–9	Линии данных. Data 0 (контакт 2) – младший бит
Ack#	O	10	Acknowledge – импульс подтверждения приема байта (запрос на прием следующего). Может использоваться для формирования запроса прерывания
Busy	O	11	Занято. Прием данных возможен только при низком уровне сигнала
PaperEnd	O	12	Высокий уровень сигнализирует о конце бумаги
Select	O	13	Сигнализирует о включении принтера (обычно в принтере соединяется резистором с цепью +5 В)
Auto LF#	I	14	Автоматический перевод строки. При низком уровне принтер, получив символ *CR* (Carriage Return – возврат каретки), автоматически выполняет и функцию *LF* (Line Feed – перевод строки)
Error#	O	32	Ошибка: конец бумаги, состояние OFF-Line или внутренняя ошибка принтера
Init#	I	31	Инициализация (сброс в режим параметров умолчания, возврат к началу строки)
Select In#	I	36	Выбор принтера (низким уровнем). При высоком уровне принтер не воспринимает остальные сигналы интерфейса
GND	–	19–30, 33	Общий провод интерфейса



Рис. 1.1. Передача данных по протоколу Centronics

Таблица 1.2. Разъем стандартного LPT-порта

Контакт DB-25S	Номер провода в кабеле	Назначение		
		I/O*	Reg. Bit**	Сигнал
1	1	O/I	CR.0\	Strobe#
2	3	O(I)	DR.0	Data 0
3	5	O(I)	DR.1	Data 1
4	7	O(I)	DR.2	Data 2
5	9	O(I)	DR.3	Data 3
6	11	O(I)	DR.4	Data 4
7	13	O(I)	DR.5	Data 5
8	15	O(I)	DR.6	Data 6
9	17	O(I)	DR.7	Data 7
10	19	I***	SR.6	Ack#
11	21	I	SR.7\	Busy
12	23	I	SR.5	PaperEnd
13	25	I	SR.4	Select
14	2	O/I	CR.1\	Auto LF#
15	4	I	SR.3	Error#
16	6	O/I	CR.2	Init#
17	8	O/I	CR.3\	Select In#
18—25	10, 12, 14, 16, 18, 20, 22, 24, 26	–	–	–

* I/O задает направление передачи (вход/выход) сигнала порта. O/I обозначает выходные линии, состояние которых считывается при чтении из портов вывода; (I) — выходные линии, состояние которых может быть считано только при особых условиях (см. ниже).

** Символом «\» отмечены инвертированные сигналы (1 в регистре соответствует низкому уровню линии).

*** Вход Ack# соединен резистором (10 кОм) с питанием +5 В.

1.1.2. Традиционный LPT-порт

Адаптер параллельного интерфейса представляет собой набор регистров, расположенных в пространстве ввода/вывода. Регистры порта адресуются относительно базового адреса порта, стандартными значениями которого являются 3BCh, 378h и 278h. Порт может использовать линию запроса аппаратного прерывания, обычно *IRQ7* или *IRQ5*. Порт имеет *внешнюю* 8-битную *шину данных*, 5-битную *шину сигналов состояния* и 4-битную *шину управляющих сигналов*.

BIOS поддерживает до четырех (иногда до трех) LPT-портов (LPT1–LPT4) своим сервисом — прерыванием *INT 17h*, обеспечивающим через них связь с принтером по интерфейсу *Centronics*. Этим сервисом BIOS осуществляет вывод символа (по опросу готовности, не используя аппаратных прерываний), инициализацию интерфейса и принтера, а также опрос состояния принтера.

Стандартный порт имеет *три 8-битных регистра*, расположенных по соседним адресам в пространстве ввода/вывода, начиная с базового адреса порта (*BASE*).

Data Register (DR) — *регистр данных*, адрес=*BASE*. Данные, записанные в этот порт, *выводятся* на выходные линии интерфейса. Данные, считанные из этого регистра, в зависимости от схемотехники адаптера соответствуют либо ранее записанным данным, либо сигналам на тех же линиях, что не всегда одно и то же. Если в порт записать байт с единицами во всех разрядах, а на выходные линии интерфейса через микросхемы с выходом типа «открытый коллектор» подать какой-либо код (или соединить ключами какие-то линии со схемной землей), то этот код может быть считан из того же регистра данных. Таким образом на многих старых моделях адаптеров можно реализовать порт ввода дискретных сигналов, однако выходным цепям передатчика информации придется «бороться» с выходным током логической единицы выходных буферов адаптера. Схемотехника ТТЛ такие решения не запрещает, но если внешнее устройство выполнено на микросхемах КМОП, их мощности может не хватить для «победы» в этом шинном конфликте. Однако современ-

ные адаптеры часто имеют в выходной цепи согласующий резистор с сопротивлением до 50 Ом. Выходной ток короткого замыкания выхода на землю обычно не превышает 30 мА. Простой расчет показывает, что в случае короткого замыкания контакта разъема на землю при выводе «единицы» на этом резисторе падает напряжение 1,5 В, что входной схемой приемника будет воспринято как «единица». Так что такой способ ввода не будет работать на всех компьютерах.

На некоторых адаптерах портов выходной буфер отключается перемычкой на плате. Тогда порт превращается в обыкновенный порт ввода.

Status Register (SR) — регистр состояния; представляет собой *5-битный порт ввода* сигналов состояния принтера (биты *SR.4–SR.7*), адрес=*BASE+1*. Бит *SR.7* инвертируется — низкому уровню сигнала соответствует единичное значение бита в регистре, и наоборот.

Назначение бит регистра состояния (в скобках даны номера контактов разъема):

- *SR.7 — Busy* — инверсные отображения состояния линии Busy (11): при низком уровне на линии устанавливается единичное значения бита — разрешение на вывод очередного байта.
- *SR.6 — Ack* (Acknowledge) — отображения состояния линии Ack# (10).
- *SR.5 — PE* (Paper End) — отображения состояния линии Paper End (12). Единичное значение соответствует высокому уровню линии — сигналу о конце бумаги в принтере.
- *SR.4 — Select* — отображения состояния линии Select (13). Единичное значение соответствует высокому уровню линии — сигналу о включении принтера.
- *SR.3 — Error* — отображения состояния линии Error (15). Нулевое значение соответствует низкому уровню линии — сигналу о любой ошибке принтера.
- *SR.2 — PIRQ* — флаг прерывания по сигналу Ack# (только для порта PS/2). Бит обнуляется, если сигнал Ack# вызвал аппаратное прерывание. Единичное значение ус-

танавливается по аппаратному сбросу и после чтения регистра состояния.

- *SR[1:0]* — зарезервированы.

Control Register (CR) — регистр управления, адрес=*BASE+2*. Как и регистр данных, этот *4-битный порт вывода* допускает запись и чтение (биты 0–3), но его выходной буфер обычно имеет тип «открытый коллектор». Это позволяет корректно использовать линии данного регистра как входные при программировании их в высокий уровень. Биты 0, 1, 3 инвертируются.

Назначение бит регистра управления:

- *CR[7:6]* — зарезервированы.
- *CR.5 — Direction* — бит управления направлением передачи (только для портов PS/2). Запись единицы переводит порт данных в режим ввода. При чтении состояние бита не определено.
- *CR.4 — AckIntEn* (Ack Interrupt Enable) — единичное значение разрешает прерывание по спаду сигнала на линии Ack# — сигнал запроса следующего байта.
- *CR.3 — Select In* — единичное значение бита соответствует низкому уровню на выходе Select In# (17) — сигналу, разрешающему работу принтера по интерфейсу *Centronics*.
- *CR.2 — Init* — нулевое значение бита соответствует низкому уровню на выходе *Init#* (16) — сигналу аппаратного сброса принтера.
- *CR.1 — Auto LF* — единичное значение бита соответствует низкому уровню на выходе Auto LF# (14) — сигналу на автоматический перевод строки (*LF* — Line Feed) по приему байта возврата каретки (CR). Иногда сигнал и бит называют *AutoFD* или *AutoFDXT*.
- *CR.0 — Strobe* — единичное значение бита соответствует низкому уровню на выходе Strobe# (1) — сигналу стробирования выходных данных.

Запрос аппаратного прерывания (обычно IRQ7 или IRQ5) вырабатывается по отрицательному перепаду сигнала на выводе 10 разъема интерфейса (Ack#) при установке *CR.4*=1. Во

избежание ложных прерываний контакт 10 соединен резистором с шиной +5 В. Прерывание вырабатывается, когда принтер подтверждает прием предыдущего байта. Как уже было сказано, BIOS это прерывание не использует и не обслуживает.

Процедура вывода байта по интерфейсу *Centronics* включает следующие шаги (в скобках приведено требуемое количество шинных операций процессора):

- Вывод байта в регистр данных (1 цикл IOWR#).
- Ввод из регистра состояния и проверка готовности устройства (бит *SR.7* — сигнал Busy). Этот шаг зацикливается до получения готовности или до срабатывания программного тайм-аута (минимум 1 цикл IORD#).
- По получении готовности выводом в регистр управления устанавливается строб данных, а следующим выводом строб снимается (2 цикла IOWR#). Обычно, чтобы переключить только один бит (строб), регистр управления предварительно считывается, что добавляет еще один цикл IORD#.

Видно, что для вывода одного байта требуется 4–5 операций ввода/вывода с регистрами порта (в лучшем случае, когда готовность обнаружена по первому чтению регистра состояния). Отсюда вытекает главный недостаток вывода через стандартный порт — невысокая скорость обмена при значительной загрузке процессора. Порт удается разогнать до скоростей 100–150 Кбайт/с при полной загрузке процессора, что недостаточно для печати на лазерном принтере. Другой недостаток — функциональный — сложность использования в качестве порта ввода.

Стандартный порт асимметричен — при наличии 12 линий (и бит), нормально работающих на вывод, на ввод работают только 5 линий состояния. Если необходима симметричная двунаправленная связь, на всех стандартных портах работоспособен *режим полубайтного обмена — Nibble Mode*. В этом режиме, называемом также *Hewlett Packard Bitronics*, одновременно передаются 4 бита данных, пятая линия используется для квитирования. Таким образом, каждый байт передается за два цикла, а каждый цикл требует по крайней мере 5 операций ввода/вывода.

1.1.3. Расширения параллельного порта

Недостатки стандартного порта частично устраняли новые типы портов, появившиеся в компьютерах PS/2.

Двунаправленный порт 1 (*Type 1 parallel port*) — интерфейс, введенный в PS/2. Такой порт кроме стандартного режима может работать в режиме ввода или двунаправленном режиме. Протокол обмена формируется программно, а для указания направления передачи в регистр управления порта введен специальный бит *CR.5*: 0 — буфер данных работает на вывод, 1 — на ввод. Не путайте этот порт, называемый также *enhanced bi-directional*, с EPP. Данный тип порта прижился и в обычных компьютерах.

Порт с прямым доступом к памяти (*Type 3 DMA parallel port*) применялся в PS/2 моделей 57, 90, 95. Был введен для повышения пропускной способности и разгрузки процессора при выводе на принтер. Программе, работающей с портом, требовалось только задать в памяти блок данных, подлежащих выводу, а затем вывод по протоколу *Centronics* производился без участия процессора.

Позже появились другие адаптеры LPT-портов, реализующие протокол обмена *Centronics* аппаратно — *Fast Centronics*. Некоторые из них использовали FIFO-буфер данных — *Parallel Port FIFO Mode*. Не будучи стандартизованными, такие порты разных производителей требовали использования собственных специальных драйверов. Программы, использующие прямое управление регистрами стандартных портов, не умели более эффективно их использовать. Такие порты часто входили в состав мультикарт VLB. Существуют их варианты с шиной ISA, в том числе встроенные.

1.2. Стандарт IEEE 1284

Стандарт на параллельный интерфейс *IEEE 1284*, принятый в 1994 году, определяет порты *SPP, EPP* и *ECP*. Стандарт определяет 5 режимов обмена данными, метод согласования режима, физический и электрический интерфейсы. Согласно IEEE 1284, возможны следующие режимы обмена данными через параллельный порт:

- *Режим совместимости* (*Compatibility Mode*) — однонаправленный (вывод) по протоколу *Centronics*. Этот режим соответствует стандартному порту *SPP*.
- *Полубайтный режим* (*Nibble Mode*) — ввод байта в два цикла (по 4 бита), используя для приема линии состояния. Этот режим обмена может использоваться на любых адаптерах.
- *Байтный режим* (*Byte Mode*) — ввод байта целиком, используя для приема линии данных. Этот режим работает только на портах, допускающих чтение выходных данных (*Bi-Directional* или *PS/2 Type 1*).
- *Режим EPP* (Enhanced Parallel Port) (*EPP Mode*) — двунаправленный обмен данными. Управляющие сигналы интерфейса генерируются аппаратно во время цикла обращения к порту. Эффективен при работе с устройствами внешней памяти и адаптерами локальных сетей.
- *Режим ECP* (Extended Capability Port) (*ECP Mode*) — двунаправленный обмен данными с возможностью аппаратного сжатия данных по методу *RLE* (Run Length Encoding) и использования FIFO-буферов и DMA. Управляющие сигналы интерфейса генерируются аппаратно. Эффективен для принтеров и сканеров.

В компьютерах с LPT-портом на системной плате режим — SPP, EPP, ECP или их комбинация — задается в BIOS Setup. Режим совместимости полностью соответствует стандартному порту SPP. Остальные режимы подробно рассмотрены ниже.

1.2.1. Физический и электрический интерфейсы

Стандарт IEEE 1284 определяет физические характеристики приемников и передатчиков сигналов. Спецификации стандартного порта не задавали типов выходных схем, предельных значений величин нагрузочных резисторов и емкости, вносимой цепями и проводниками. На относительно невысоких скоростях обмена разброс этих параметров не вызывал проблем совместимости. Однако расширенные (функционально и по скорости передачи) режимы

требуют четких спецификаций. IEEE 1284 определяет два уровня интерфейсной совместимости. *Первый уровень* (Level I) определен для устройств медленных, но использующих смену направления передачи данных. *Второй уровень* (Level II) определен для устройств, работающих в расширенных режимах, с высокими скоростями и длинными кабелями. К *передатчикам* предъявляются следующие требования:

- Уровни сигналов без нагрузки не должны выходить за пределы −0,5... +5,5 В.
- Уровни сигналов при токе нагрузки 14 мА должны быть не ниже +2,4 В для высокого уровня (V_{OH}) и не выше +0,4 В для низкого уровня (V_{OL}) на постоянном токе.
- Выходной импеданс R_O, измеренный на разъеме, должен составлять 50±5 Ом на уровне $V_{OH}-V_{OL}$. Для обеспечения заданного импеданса используют последовательные резисторы в выходных цепях передатчика. Согласование импеданса передатчика и кабеля снижает уровень импульсных помех.
- Скорость нарастания (спада) импульса должна находиться в пределах 0,05−0,4 В/нс.

Требования к *приемникам*:

- Допустимые пиковые значения сигналов −2,0...+7,0 В.
- Пороги срабатывания должны быть не выше 2,0 В (V_{IH}) для высокого уровня и не ниже 0,8 В (V_{IL}) для низкого.
- Приемник должен иметь гистерезис в пределах 0,2...1,2 В (гистерезисом обладают специальные микросхемы — триггеры Шмитта).
- Входной ток микросхемы (втекающий и вытекающий) не должен превышать 20 мкА, входные линии соединяются с шиной питания +5 В резистором 1,2 кОм.
- Входная емкость не должна превышать 50 пФ.

Когда появилась спецификация ECP, фирма Microsoft рекомендовала применение динамических терминаторов на каждую линию интерфейса. Однако в настоящее время следуют спецификации IEEE 1284, в которой динамические

терминаторы не применяются. Рекомендованные схемы входных, выходных и двунаправленных цепей приведены на рис. 1.2.

Стандарт IEEE 1284 определяет три типа используемых *разъемов*. Типы *A* (*DB-25*) и *B* (*Centronics-36*) используются в традиционных кабелях подключения принтера, тип *C* — новый малогабаритный 36-контактный разъем.

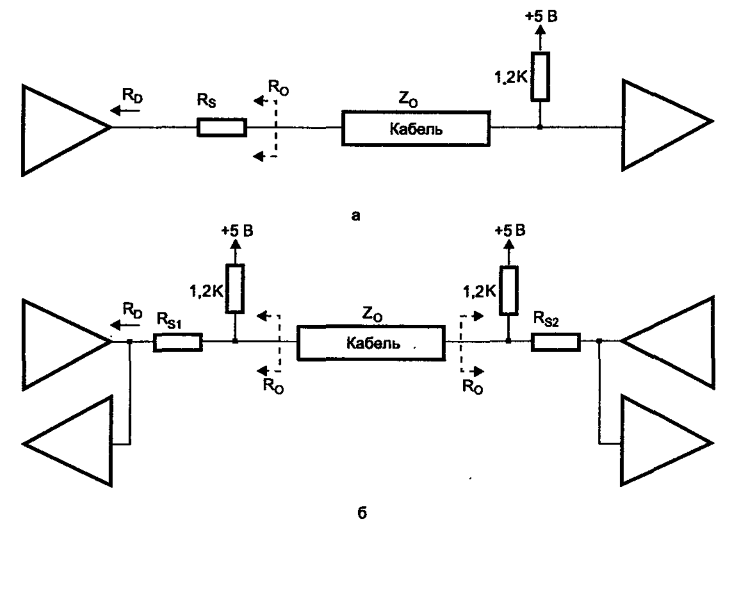


Рис. 1.2. Оконечные цепи линий интерфейса IEEE 1284: *а* — однонаправленных, *б* — двунаправленных

Традиционные интерфейсные кабели имеют от 18 до 25 проводов, в зависимости от числа проводников цепи GND. Эти проводники могут быть как перевитыми, так и нет. К экранированию кабеля жестких требований не предъявлялось. Такие кабели вряд ли будут надежно работать на скорости передачи 2 Мбайт/с и при длине более 2 м. Стандарт IEEE 1284 регламентирует *свойства кабелей*.

- Все сигнальные линии должны быть перевитыми с отдельными обратными (общими) проводами.
- Каждая пара должна иметь импеданс 62±6 Ом в частотном диапазоне 4–16 МГц.
- Уровень перекрестных помех между парами не должен превышать 10%.
- Кабель должен иметь экран (фольгу), покрывающий не менее 85% внешней поверхности. На концах кабеля экран должен быть окольцован и соединен с контактом разъема.

Кабели, удовлетворяющие этим требованиям, маркируются надписью «*IEEE Std 1284-1994 Compliant*». Они могут иметь длину до 10 метров, обозначения типов приведены в табл. 1.3.

Таблица 1.3. Типы кабелей IEEE 1284

Тип	Расшифровка	Разъем 1	Разъем 2
AMAM	Type A Male – Type A Male	A (вилка)	A (вилка)
AMAF	Type A Male – Type A Female	A (вилка)	A (розетка)
AB	Type A Male – Type B Plug (стандартный кабель к принтеру)	A (вилка)	B
AC	Type A Male – Type C Plug (новый кабель к принтеру)	A (вилка)	C
BC	Type B Plug – Type C Plug	B	C
CC	Type C Plug – Type C Plug	C	C

1.2.2. Режимы передачи данных

IEEE 1284 определяет пять *режимов обмена*, один из которых полностью соответствует стандартному выводу по протоколу *Centronics*. Стандарт определяет способ, по которому ПО может определить режим, доступный и хосту (PC), и ПУ (или присоединенному второму компьютеру). Режимы нестандартных портов, реализующих протокол обмена *Centronics* аппаратно (*Fast Centronics*, *Parallel Port FIFO Mode*),

могут и не являться режимами IEEE 1284, несмотря на наличие в них черт *EPP* и *ECP*.

При описании режимов обмена фигурируют следующие понятия:

- *Хост* — компьютер, обладающий параллельным портом.
- *ПУ* — периферийное устройство, подключаемое к этому порту.
- Ptr — в обозначениях сигналов обозначает передающее ПУ.
- *Прямой канал* — канал вывода данных от хоста в ПУ.
- *Обратный канал* — канал ввода данных в хост из ПУ.

Полубайтный режим ввода — Nibble Mode

Предназначен для двунаправленного обмена. Может работать на всех стандартных портах. Порты имеют *5 линий ввода* состояния, используя которые ПУ может посылать в хост байт тетрадами (nibble — полубайт, 4 бита) за два приема. Сигнал Ack#, вызывающий прерывание, которое может использоваться в данном режиме, соответствует биту 6 регистра состояния, что усложняет программные манипуляции с битами при сборке байта. Сигналы порта приведены в табл. 1.4, временные диаграммы — на рис. 1.3.

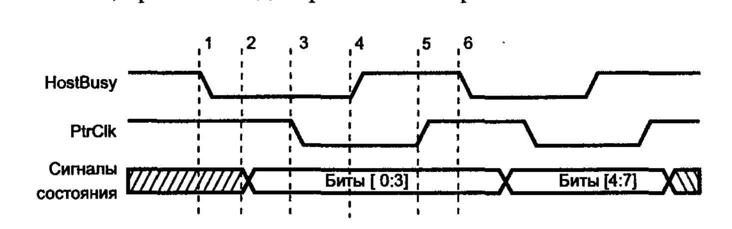


Рис. 1.3. Прием данных в полубайтном режиме

Прием байта данных в полубайтном режиме состоит из следующих фаз:

1. Хост сигнализирует о готовности приема данных установкой низкого уровня на линии HostBusy.
2. ПУ в ответ помещает тетраду на входные линии состояния.

3. ПУ сигнализирует о готовности тетрады установкой низкого уровня на линии PtrClk.
4. Хост устанавливает высокий уровень на линии HostBusy, указывая на занятость приемом и обработкой тетрады.
5. ПУ отвечает установкой высокого уровня на линии PtrClk.
6. Шаги 1–5 повторяются для второй тетрады.

Таблица 1.4. Сигналы LPT-порта в полубайтном режиме ввода

Контакт	Сигнал SPP	I/O	Описание
14	AutoFeed#	O	HostBusy — сигнал квитирования. Низкий уровень означает готовность к приему тетрады, высокий подтверждает прием тетрады
17	SelectIn#	O	Высокий уровень указывает на обмен в режиме IEEE 1284 (в режиме SPP уровень низкий)
10	Ack#	I	PtrClk. Низкий уровень означает готовность тетрады, высокий — ответ на сигнал HostBusy
11	Busy	I	Прием бита данных 3, затем бита 7
12	PE	I	Прием бита данных 2, затем бита 6
13	Select	I	Прием бита данных 1, затем бита 5
15	Error#	I	Прием бита данных 0, затем бита 4

Полубайтный режим сильно нагружает процессор, и поднять скорость обмена выше 50 Кбайт/с не удается. Безусловное его преимущество в том, что он работает *на всех портах*. Его применяют в тех случаях, когда поток данных невелик (например, для связи с принтерами). Однако при связи с адаптерами локальных сетей, внешними дисковыми накопителями и CD-ROM прием больших объемов данных требует изрядного терпения со стороны пользователя.

Двунаправленный байтный режим – Byte Mode

В данном режиме данные принимаются с использованием двунаправленного порта, у которого выходной буфер дан-

ных может отключаться установкой бита *CR.5*=1. Как и предыдущие, режим является программно-управляемым — все сигналы квитирования анализируются и устанавливаются драйвером. Сигналы порта описаны в табл. 1.5, временные диаграммы — на рис. 1.4.

Таблица 1.5. Сигналы LPT-порта в байтном режиме ввода/вывода

Контакт	Сигнал SPP	Имя в байтном режиме	I/O	Описание
1	Strobe#	HostClk	O	Импульс (низкого уровня) подтверждает прием байта в конце каждого цикла
14	AutoFeed#	HostBusy	O	Сигиал квитирования. Низкий уровень означает готовность хоста принять байт; высокий уровень устанавливается по приему байта
17	SelectIn#	1284Active	O	Высокий уровень указывает на обмен в режиме IEEE 1284 (в режиме SPP уровень низкий)
16	Init#	Init#	O	Не используется; установлен высокий уровень
10	Ack#	PtrClk	I	Устанавливается в низкий уровень для индикации действительности данных на линиях Data [0:7]. Низкий уровень устанавливается в ответ на сигнал HostBusy
11	Busy	PtrBusy	I	Состояние занятости прямого канала
12	PE	AckDataReq*	I	Устанавливается ПУ для указания на наличие обратного канала передачи
13	Select	Xflag*	I	Флаг расширяемости
15	Error#	DataAvail#*	I	Устанавливается ПУ для указания на наличие обратного канала передачи
2–9	Data [0:7]	Data [0:7]	I/O	Двунаправленный (прямой и обратный) канал данных

* Сигналы действуют в последовательности согласования (см. далее).

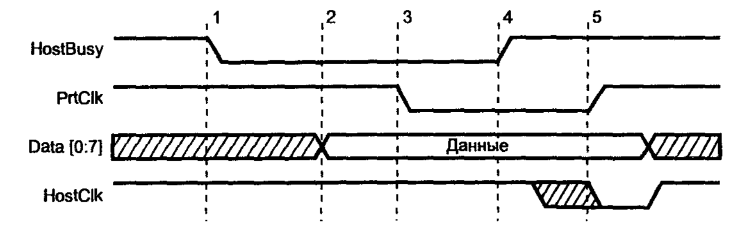


Рис. 1.4. Прием данных в байтном режиме

Фазы приема байта данных:

1. Хост сигнализирует о готовности приема данных установкой низкого уровня на линии HostBusy.
2. ПУ в ответ помещает байт данных на линии Data [0:7].
3. ПУ сигнализирует о действительности байта установкой низкого уровня на линии PtrClk.
4. Хост устанавливает высокий уровень на линии HostBusy, указывая на занятость приемом и обработкой байта.
5. ПУ отвечает установкой высокого уровня на линии PtrClk.
6. Хост подтверждает прием байта импульсом HostClk.
7. Шаги 1–6 повторяются для каждого следующего байта.

Побайтный режим позволяет поднять скорость обратного канала до скорости прямого канала в стандартном режиме. Однако он может работать *только на двунаправленных портах.*

Режим EPP

Протокол *EPP* (Enhanced Parallel Port — улучшенный параллельный порт) был разработан компаниями Intel, Xircom и Zenith Data Systems задолго до принятия IEEE 1284. Он предназначен для повышения производительности обмена по параллельному порту. *EPP* был реализован в чипсете Intel 386SL (микросхема 82360) и впоследствии принят множеством компаний как дополнительный протокол параллельного порта. Версии протокола, реализованные до принятия IEEE 1284, отличаются от нынешнего стандарта (см. далее).

Протокол *EPP* обеспечивает четыре типа циклов обмена:

- запись данных;
- чтение данных;
- запись адреса;
- чтение адреса.

Назначение циклов записи и чтения данных очевидно. Адресные циклы используются для передачи адресной, канальной и управляющей информации. Циклы обмена данными отличаются от адресных циклов применяемыми стробирующими сигналами. Назначение сигналов порта *EPP* и их связь с сигналами *SPP* объясняются в табл. 1.6.

Таблица 1.6. Сигналы LPT-порта в режиме ввода/вывода EPP

Контакт	Сигнал SPP	Имя в EPP	I/O	Описание
1	Strobe#	Write#	O	Низкий уровень — цикл записи, высокий — цикл чтения
14	AutoLF#	DataStb#	O	Строб данных. Низкий уровень устанавливается в циклах передачи данных
17	SelectIn#	AddrStb#	O	Строб адреса. Низкий уровень устанавливается в адресных циклах
16	Init#	Reset#	O	Сброс ПУ (низким уровнем)
10	Ack#	INTR#	I	Прерывание от ПУ
11	Busy	Wait#	I	Сигнал квитирования. Низкий уровень разрешает начало цикла (установку строба в низкий уровень), переход в высокий — разрешает завершение цикла (снятие строба)
2–9	Data [0:7]	AD[0:7]	I/O	Двунаправленная шина адреса/данных
12	PaperEnd	AckDataReq*	I	Используется по усмотрению разработчика периферии
13	Select	Xflag*	I	Используется по усмотрению разработчика периферии
15	Error#	DataAvail#*	I	Используется по усмотрению разработчика периферии

* Сигналы действуют в последовательности согласования (см. далее).

EPP-порт имеет *расширенный набор регистров* (табл. 1.7), который занимает в пространстве ввода/вывода 5–8 смежных байт.

Таблица 1.7. Регистры EPP-порта

Имя регистра	Смещение	Режим	R/W	Описание
SPP Data Port	+0	*SPP/EPP*	W	*Регистр данных SPP*
SPP Status Port	+1	*SPP/EPP*	R	*Регистр состояния SPP*
SPP Control Port	+2	*SPP/EPP*	W	*Регистр управления SPP*
EPP Address Port	+3	*EPP*	R/W	*Регистр адреса EPP*. Чтение или запись в него генерирует связанный цикл чтения или записи адреса *EPP*
EPP Data Port	+4	*EPP*	R/W	*Регистр данных EPP*. Чтение (запись) генерирует связанный цикл чтения (записи) данных *EPP*
Not Defined	+5...+7	*EPP*	N/A	В некоторых контроллерах могут использоваться для 16-/32-битных операций ввода/вывода

В отличие от программно-управляемых режимов, описанных ранее, внешние сигналы *EPP*-порта для каждого цикла обмена формируются аппаратно по одной операции записи или чтения в регистр порта. На рис. 1.5 приведена диаграмма *цикла записи* данных, иллюстрирующая внешний цикл обмена, *вложенный* в цикл записи системной шины процессора (иногда эти циклы называют *связанными*). Адресный цикл записи отличается от цикла данных только стробом внешнего интерфейса.

Цикл записи данных состоит из следующих фаз:

1. Программа выполняет цикл вывода (IOWR#) в порт 4 (*EPP Data Port*).
2. Адаптер устанавливает сигнал Write# (низкий уровень), и данные помещаются на выходную шину LPT-порта.

3. При низком уровне Wait# устанавливается строб данных.
4. Порт ждет подтверждения от ПУ (перевода Wait# в высокий уровень).
5. Снимается строб данных — внешний *EPP*-цикл завершается.
6. Завершается процессорный цикл вывода.
7. ПУ устанавливает низкий уровень Wait#, указывая на возможность начала следующего цикла.

Рис. 1.5. Цикл записи данных EPP

Пример адресного *цикла чтения* приведен на рис. 1.6. Цикл чтения данных отличается только применением другого стробирующего сигнала.

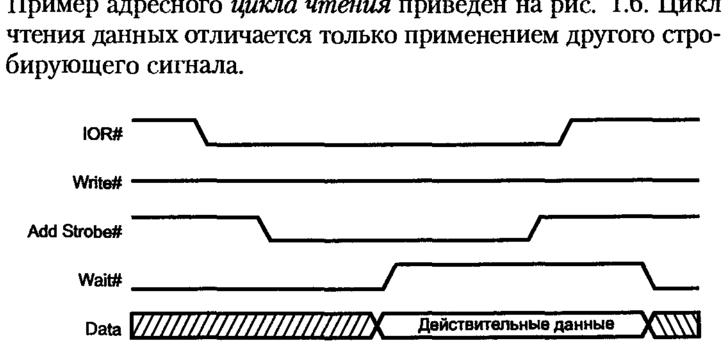


Рис. 1.6. Адресный цикл чтения EPP

Главной отличительной чертой *EPP* является выполнение внешней передачи во время одного процессорного цикла вво-

да/вывода. Это позволяет достигать высоких скоростей обмена (0,5...2 Мбайт/с). ПУ, подключенное к параллельному порту *EPP*, может работать со скоростью устройства, подключаемого через слот ISA. Протокол блокированного квитирования (*interlocked handshakes*) позволяет автоматически настраиваться на скорость обмена, доступную и хосту, и ПУ. ПУ может регулировать длительность всех фаз обмена с помощью всего лишь одного сигнала Wait#. Протокол автоматически подстраивается под длину кабеля — вносимые задержки приведут только к удлинению цикла. Поскольку кабели, соответствующие IEEE 1284 (см. выше), имеют одинаковые волновые свойства для разных линий, нарушения передачи, связанного с «состязаниями» сигналов, происходить не должно. При подключении сетевых адаптеров или внешних дисков к *EPP*-порту можно наблюдать непривычное явление: снижение производительности по мере удлинения интерфейсного кабеля.

Естественно, ПУ не должно «подвешивать» процессор на шинном цикле обмена. Это гарантирует механизм тайм-аутов PC, который принудительно завершает любой цикл обмена, длящийся более 15 мкс. В ряде реализаций *EPP* за тайм-аутом интерфейса следит сам адаптер — если ПУ не отвечает в течение определенного времени (5 мкс), цикл прекращается и в дополнительном (не стандартизованном) регистре состояния адаптера фиксируется ошибка.

Устройства с интерфейсом *EPP*, разработанные до принятия IEEE 1284, отличаются началом цикла: строб DataStb# или AddrStb# устанавливается независимо от состояния WAIT#. Это означает, что ПУ не может задержать начало следующего цикла (хотя может растянуть его на требуемое время). Такая спецификация называется *EPP 1.7* (предложена Xircom). Именно она применялась в контроллере 82360. Периферия, совместимая с *EPP 1.7*, будет нормально работать и с контроллером *EPP 1284*, но периферия в стандарте *EPP 1284* может отказаться работать с контроллером *EPP 1.7*.

С *программной* точки зрения контроллер *EPP*-порта выглядит просто (см. табл. 1.7). К трем регистрам стандартного порта, имеющим смещение 0, 1 и 2 относительно базового

адреса порта, добавлены два регистра (*EPP Address Port* и *EPP Data Port*), чтение и запись в которые вызывает генерацию связанных внешних циклов.

Назначение регистров стандартного порта сохранено для совместимости *EPP*-порта с ПУ и ПО, рассчитанными на применение программно-управляемого обмена. Поскольку сигналы квитирования адаптером вырабатываются аппаратно, при записи в регистр управления *CR* биты 0, 1 и 3, соответствующие сигналам Strobe#, AutoFeed# и SelectIn#, должны иметь нулевые значения. Программное вмешательство могло бы нарушить последовательность квитирования. Некоторые адаптеры имеют специальные средства защиты (*EPP Protect*), при включении которых программная модификация этих бит блокируется.

Использование регистра данных *EPP* позволяет осуществлять передачу блока данных с помощью одной инструкции *REP INSB* или *REP OUTSB*. Некоторые адаптеры допускают *16-/32-битное обращение* к регистру данных *EPP*. При этом адаптер просто дешифрует адрес со смещением в диапазоне 4–7 как адрес регистра данных *EPP*, но процессору сообщает о разрядности 8 бит. Тогда 16- или 32-битное обращение по адресу регистра данных *EPP* приведет к автоматической генерации двух или четырех шинных циклов по нарастающим адресам, начиная со смещения 4. Эти циклы будут выполняться быстрее, чем то же количество одиночных циклов. Таким образом обеспечивается производительность до 2 Мбайт/с, достаточная для адаптеров локальных сетей, внешних дисков, стриммеров и CD-ROM. Адресные циклы *EPP* всегда выполняются только в однобайтном режиме.

Важной чертой *EPP* является то, что обращение процессора к ПУ осуществляется в реальном времени — нет буферизации. Драйвер способен отслеживать состояние и подавать команды в точно известные моменты времени. Циклы чтения и записи могут чередоваться в произвольном порядке или идти блоками. Такой тип обмена удобен для *регистро-ориентированных ПУ* или ПУ, работающих в *реальном времени*, — сетевых адаптеров, устройств сбора информации и управления и т. п.

Режим ECP

Протокол *ECP* (Extended Capability Port — порт с расширенными возможностями) был предложен Hewlett Packard и Microsoft для связи с ПУ типа принтеров или сканеров. Как и *EPP*, данный протокол обеспечивает высокопроизводительный двунаправленный обмен данными хоста с ПУ.

Протокол *ECP* в обоих направлениях обеспечивает два типа циклов:

- циклы записи и чтения данных;
- командные циклы записи и чтения.

Командные циклы подразделяются на два типа: передачу канальных адресов и передачу счетчика *RLC* (Run-Length Count).

В отличие от *EPP*, вместе с протоколом *ECP* сразу появился стандарт на программную (регистровую) модель его адаптера, изложенный в документе «The IEEE 1284 Extended Capabilities Port Protocol and ISA Interface Standard» компании Microsoft. Этот документ определяет свойства протокола, не заданные стандартом IEEE 1284:

- компрессию данных хост-адаптером по методу RLE;
- буферизацию FIFO для прямого и обратного каналов;
- применение DMA и программного ввода/вывода.

Компрессия в реальном времени по методу *RLE* (Run-Length Encoding) позволяет достичь коэффициента сжатия 64:1 при передаче растровых изображений, которые имеют длинные строки повторяющихся байт. Компрессию можно использовать, только если ее поддерживает и хост, и ПУ.

Канальная адресация ECP применяется для адресации множества логических устройств, входящих в одно физическое. Например, в комбинированном устройстве факс/принтер/модем, подключаемом только к одному параллельному порту, возможен одновременный прием факса и печать на принтере. В режиме *SPP*, если принтер установит сигнал занятости, канал будет занят данными, пока принтер их не примет. В режиме *ECP* программный драйвер просто адресуется к другому *логическому каналу* того же порта.

Протокол *ECP* переопределяет сигналы *SPP* (табл. 1.8).

Таблица 1.8. Сигналы LPT-порта в режиме ввода/вывода ECP

Контакт	Сигная SPP	Имя в ECP	I/O	Описание
1	Strobe#	HostClk	O	Используется в паре с PeriphAck для передачи в прямом направлении (вывод)
14	AutoLF#	HostAck	O	Указывает тип цикла (команда/данные) при передаче в прямом направлении. Используется в паре с PeriphClk для передачи в обратном направлении
17	SelectIn#	1284Active	O	Высокий уровень указывает на обмен в режиме IEEE 1284 (в режиме *SPP* уровень низкий)
16	Init#	ReverseRequest#	O	Низкий уровень переключает канал на передачу в обратном направлении
10	Ack#	PeriphClk	I	Используется в паре с HostAck для передачи в обратном направлении
11	Busy	PeriphAck	I	Используется в паре с HostClk для передачи в обратном направлении. Индицирует тип команда/данные при передаче в обратном направлении
12	PaperEnd	AckReverse#	I	Переводится в низкий уровень как подтверждение сигналу ReverseRequest#
13	Select	Xflag*	I	Флаг расширяемости
15	Error#	PeriphRequest#*	I	Устанавливается ПУ для указания на доступность (наличие) обратного канала передачи*
2–9	Data [0:7]	Data [0:7]	I/O	Двунаправленный канал данных

* Сигналы действуют в последовательности согласования (см. далее).

Адаптер *ECP* тоже генерирует внешние протокольные сигналы квитирования аппаратно, но его работа существенно отличается от режима *EPP*.

На рис. 1.7а приведена диаграмма двух циклов прямой передачи: за циклом данных следует командный цикл. *Тип цикла* задается уровнем на линии HostAck: в цикле данных — высокий, в командном цикле — низкий. В командном цикле байт может содержать *канальный адрес* или счетчик RLE. Отличительным признаком является бит 7 (старший): если он нулевой, то биты 0–6 содержат счетчик *RLE* (0–127), если единичный — то канальный адрес. На рис. 1.7б показана пара циклов обратной передачи.

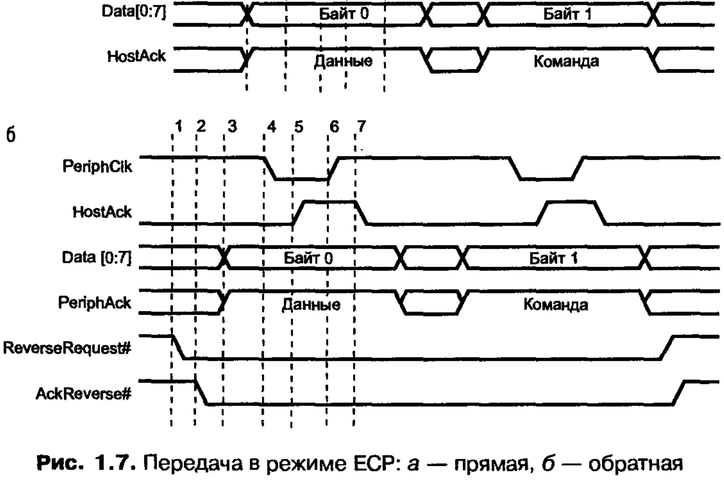


Рис. 1.7. Передача в режиме ECP: *а* — прямая, *б* — обратная

В отличие от диаграмм обмена *EPP*, на рис. 1.7 не приведены сигналы циклов системной шины процессора. В данном режиме обмен программы с ПУ разбивается на два относительно независимых процесса, которые связаны через FIFO-буфер. Обмен драйвера с FIFO-буфером может осуществляться с использованием как DMA, так и программного

ввода-вывода. Обмен ПУ с буфером аппаратно выполняет адаптер *ECP*. Драйвер в режиме *ECP* не имеет информации о точном состоянии процесса обмена, но здесь обычно важно только то, завершен он или нет. *Прямая передача* данных на внешнем интерфейсе состоит из следующих шагов:

1. Хост помещает данные на шину канала и устанавливает признак цикла данных (высокий уровень) или команды (низкий уровень) на линии HostAck.
2. Хост устанавливает низкий уровень на линии HostClk, указывая на действительность данных.
3. ПУ отвечает установкой высокого уровня на линии PeriphAck.
4. Хост устанавливает высокий уровень линии HostClk, и этот перепад может использоваться для фиксации данных в ПУ.
5. ПУ устанавливает низкий уровень на линии PeriphAck для указания на готовность к приему следующего байта.

Поскольку передача в *ECP* происходит через FIFO-буферы, которые могут присутствовать на обеих сторонах интерфейса, важно понимать, на каком этапе данные можно считать переданными. Данные считаются *переданными* на шаге 4, когда линия HostClk переходит в высокий уровень. В этот момент модифицируются счетчики переданных и принятых байт. В протоколе *ECP* есть условия, вызывающие прекращение обмена между шагами 3 и 4. Тогда эти данные не должны рассматриваться как переданные.

Из рис. 1.7 видно и другое отличие *ECP* от *EPP*. Протокол *EPP* позволяет драйверу чередовать циклы прямой и обратной передачи, не запрашивая подтверждения на смену направления. В *ECP смена направления* должна быть согласована: хост запрашивает реверс установкой ReverseRequest#, после чего он должен дождаться подтверждения сигналом AckReverse#. Поскольку предыдущий цикл мог выполняться по прямому доступу, драйвер должен дождаться завершения прямого доступа или прервать его, выгрузить буфер FIFO, определив точное значение счетчика переданных байт, и только после этого запрашивать реверс.

Обратная передача данных состоит из следующих шагов:

1. Хост запрашивает изменение направления канала, устанавливая низкий уровень на линии ReverseRequest#.
2. ПУ разрешает смену направления установкой низкого уровня на линии AckReverse#.
3. ПУ помещает данные на шину канала и устанавливает признак цикла данных (высокий уровень) или команды (низкий уровень) на линии PeriphAck.
4. ПУ устанавливает низкий уровень на линии PeriphClk, указывая на действительность данных.
5. Хост отвечает установкой высокого уровня на линии HostAck.
6. ПУ устанавливает высокий уровень линии PeriphClk; этот перепад может использоваться для фиксации данных хостом.
7. Хост устанавливает низкий уровень на линии HostAck для указания на готовность к приему следующего байта.

Режимы и регистры ECP-порта

Программный интерфейс и регистры *ECP* для адаптеров IEEE 1284 определяет спецификация Microsoft. Определены режимы (табл. 1.9), в которых может функционировать адаптер. Они задаются полем *Mode* регистра *ECR* (биты *[7:5]*).

Регистровая модель адаптера *ECP* (табл. 1.10) использует свойства архитектуры стандартной шины и адаптеров ISA — для дешифрации адресов портов ввода/вывода задействуются только 10 младших линий шины адреса. Поэтому, например, обращения по адресам *Port*, *Port+400h*, *Port+800h*... будут восприниматься как обращения к адресу *Port*, лежащему в диапазоне 0–3FFh. Современные PC и адаптеры декодируют большее количество адресных бит, поэтому обращения по адресам 0378h и 0778h будет адресованы двум различным регистрам. Помещение дополнительных регистров *ECP* «за спину» регистров стандартного порта (смещение 400–402h) преследует две цели. Во-первых, эти адреса никогда не использовались традиционными адаптерами и их драйверами, и их применение в *ECP* не приведет к сужению доступного

адресного пространства ввода/вывода. Во-вторых, этим обеспечивается совместимость со старыми адаптерами на уровне режимов 000–001 и возможность определения присутствия *ECP*-адаптера через попытку обращения к его расширенным регистрам.

Таблица 1.9. Режимы ECP-порта

Режим	Название	Описание
000	*SPP mode*	Стандартный (традиционный) режим
001	*Bi-directional mode*	Двунаправленный порт (тип 1 для PS/2)
010	*Fast Centronics*	Однонаправленный с использованием FIFO и DMA
011	*ECP Parallel Port mode*	*ECP*
100	*EPP Parallel Port mode**	Перевод в режим *EPP*
101	Зарезервировано	—
110	*Test mode*	Тестирование работы FIFO и прерываний
111	*Configuration mode*	Доступ к конфигурационным регистрам

* Этот режим не входит в спецификацию Microsoft, но трактуется как EPP контроллером SMC FDC37C665/666 и многими другими.

Каждому режиму *ECP* соответствуют (и доступны) свои функциональные регистры. Переключение режимов осуществляется записью в регистр ***ECR***. «Дежурными» режимами, включаемыми по умолчанию, являются 000 или 001. В любом из них работает полубайтный режим ввода. Из этих режимов всегда можно переключиться в любой другой, но из старших режимов (010–111) переключение возможно только в 000 или 001. Для корректной работы интерфейса перед выходом из старших режимов необходимо дождаться завершения обмена по прямому доступу и очистки FIFO-буфера.

В ***режиме 000 (SPP)*** порт работает как стандартный однонаправленный программно-управляемый *SPP*.

В ***режиме 001 (Bi-Di PS/2)*** порт работает как двунаправленный порт PS/2 типа 1. От режима 000 отличается возможностью реверса канала данных по биту ***CR.5***.

Режим 010 *(Fast Centronics)* предназначен только для высокопроизводительного вывода через FIFO-буфер с использованием DMA. Сигналы квитирования по протоколу *Centronics* вырабатываются аппаратно. Сигнал запроса прерывания вырабатывается по состоянию FIFO-буфера, но не по сигналу Ack# (запрос одиночного байта «не интересует» драйвер быстрого блочного вывода).

Режим 011 является собственно режимом *ECP*, описанным ранее. Поток данных и команд, передаваемых в ПУ, помещается в FIFO-буфер через регистры *ECPDFIFO* и *ECPAFIFO* соответственно. Из FIFO они выводятся с соответствующим признаком цикла (состояние линии HostAck). Принимаемый поток данных от ПУ извлекается из FIFO-буфера через регистр *ECPDFIFO*. Получение адреса в командном цикле от ПУ не предусматривается. Обмен с регистром *ECPDFIFO* может производиться и по каналу DMA.

Компрессия по методу RLE при передаче выполняется программно. Для передачи подряд более двух одинаковых байт данных в регистр *ECPAFIFO* записывается байт, у которого младшие 7 бит содержат счетчик *RLC* (значение RLC=127 соответствует 128 повторам), а старший бит нулевой. После этого в *ECPDFIFO* записывается сам байт. Отсюда очевидно, что вывод данных с одновременным использованием компрессии и DMA невозможен. Принимая эту пару байт (командный байт и байт данных), ПУ осуществляет декомпрессию. При приеме потока от ПУ адаптер *ECP* декомпрессию осуществляет аппаратно и в FIFO-буфер помещает уже декомпрессированные данные.

Режим 100 *(EPP)* — один из способов включения режима *EPP*.

Режим 110 *(Test Mode)* предназначен для тестирования взаимодействия FIFO и прерываний. Данные могут передаваться в/из регистра *TFIFO* с помощью DMA или программным способом. На внешний интерфейс обмен не воздействует. Адаптер отрабатывает операции вхолостую на максимальной скорости интерфейса (как будто сигналы квитирования приходят без задержек). Адаптер следит за состоянием буфера и по мере необходимости вырабатывает сигналы запроса прерывания. Таким образом программа

может определить максимальную пропускную способность канала.

Режим 111 (Configuration mode) предназначен для доступа к конфигурационным регистрам. Выделение режима защищает адаптер и протокол от некорректных изменений конфигурации в процессе обмена.

Таблица 1.10. Регистры ECP

Смещение	Имя	R/W	Режимы ECP*	Название
000	*DR*	R/W	000-001	*Data Register*
000	*ECPAFIFO*	R/W	011	*ECP Address FIFO*
001	*SR*	R/W	Все	*Status Register*
002	*CR*	R/W	Все	*Control Register*
400	*SDFIFO*	R/W	010	*Parallel Port Data FIFO*
400	*ECPDFIFO*	R/W	011	*ECP Data FIFO*
400	*TFIFO*	R/W	110	*Test FIFO*
400	*CNFGA*	R	111	*Configuration Register A*
401	*CNFGB*	R/W	111	*Configuration Register B*
402	*ECR*	R/W	Все	*Extended Control Register*

* Регистры доступны только в указанных режимах (режим задается битами 7–5 регистра ECR).

Регистр данных DR используется для передачи данных только в программно-управляемых режимах (000 и 001).

Регистр состояния SR передает значение сигналов на соответствующих линиях (как в *SPP*).

Регистр управления CR имеет назначение бит, совпадающее с *SPP*. В режимах 010, 011 запись в биты 0, 1 (сигналы AutoLF# и Strobe#) игнорируется.

Регистр ECPAFIFO служит для помещения информации командных циклов (канального адреса или счетчика *RLE*, в зависимости от бита 7) в FIFO-буфер. Из буфера информация будет выдана в командном цикле вывода.

Регистр SDFIFO используется для передачи данных в режиме 010. Данные, записанные в регистр (или посланные по каналу DMA), передаются через буфер FIFO по реализованному аппаратно протоколу *Centronics*. При этом должно быть задано прямое направление передачи (бит *CR.5*=0).

Регистр DFIFO используется для обмена данными в режиме 011 (*ECP*). Данные, записанные в регистр или считанные из него (или переданные по каналу DMA), передаются через буфер FIFO по протоколу *ECP*.

Регистр TFIFO обеспечивает механизм тестирования FIFO-буфера в режиме 110.

Регистр ECPCFGA позволяет считывать информацию об адаптере (идентификационный код в битах [7:4].

Регистр ECPCFGB хранит информацию, необходимую драйверу. Запись в регистр не влияет на работу порта.

Регистр ECR — главный управляющий регистр *ECP*.

Назначение бит регистра *ECR*:

- *ECR[7:5] — ECP MODE* — задает режим *ECP*.
- *ECR.4 — ERRINTREN#* (Error Interrupt Disable) — запрещает прерывания по сигналу Error# (при нулевом значении бита по отрицательному перепаду на этой линии вырабатывается запрос прерывания).
- *ECR.3 — DMAEN* (DMA Enable) — разрешает обмен по каналу DMA.
- *ECR.2 — SERVICEINTR* (Service Interrupt) — запрещает сервисные прерывания, которые вырабатываются по окончании цикла DMA (если он разрешен), по порогу заполнения/опустошения FIFO-буфера (если не используется DMA) и по ошибке переполнения буфера сверху или снизу.
- *ECR.1 — FIFOFS* (FIFO Full Status) — сигнализирует о заполнении буфера; при FIFOFS=1 в буфере нет ни одного свободного байта.
- *ECR.0 — FIFOES* (FIFO Empty Status) — указывает на полное опустошение буфера; комбинация *FIFOFS*=*FIFOES*=1 означает ошибку работы с FIFO (переполнение сверху или снизу).

Когда порт находится в стандартном или двунаправленном режимах (000 или 001), первые три регистра полностью совпадают с регистрами стандартного порта. Так обеспечивается совместимость драйвера со старыми адаптерами и старых драйверов с новыми адаптерами.

По интерфейсу с программой *ECP*-порт напоминает *EPP*: после установки режима (записи кода в регистр *ECR*) обмен данными с устройством сводится к чтению или записи в соответствующие регистры. За состоянием FIFO-буфера наблюдают либо по регистру *ECR*, либо по обслуживанию сервисных прерываний от порта. Весь протокол квитирования генерируется адаптером аппаратно. Обмен данными с *ECP*-портом (кроме явного программного) возможен и по прямому доступу к памяти (каналу DMA), что эффективно при передаче больших блоков данных.

1.2.3. Согласование режимов IEEE 1284

ПУ в стандарте IEEE 1284 обычно не требуют от контроллера реализации всех режимов, предусмотренных стандартом. Для определения режимов и методов управления конкретным устройством стандарт предусматривает *последовательность согласования* (*negotiation sequence*). Последовательность построена так, что старые устройства, не рассчитанные на применение IEEE 1284, на нее не ответят, и контроллер останется в стандартном режиме. Периферия IEEE 1284 может сообщить о своих возможностях, и контроллер установит режим, удовлетворяющий и хост, и ПУ.

Во время фазы согласования контроллер выставляет на линии данных *байт расширяемости* (*extensibility byte*), запрашивая подтверждение на перевод интерфейса в требуемый режим или прием идентификатора ПУ (табл. 1.11). Идентификатор передается контроллеру в запрошенном режиме (любой режим обратного канала, кроме *EPP*). ПУ использует сигнал Xflag (Select в терминах *SPP*) для подтверждения запрошенного режима обратного канала, кроме полубайтного. Полубайтный режим поддерживается всеми устройствами IEEE 1284. Бит *Extensibility Link request* послужит для определения дополнительных режимов в будущих расширениях стандарта.

Таблица 1.11. Биты в байте расширяемости

Бит	Описание	Допустимые комбинации бит [7:0]
7	Request Extensibility Link — зарезервирован	1000 0000
6	Запрос режима *EPP*	0100 0000
5	Запрос режима *ECP* с RLE	0011 0000
4	Запрос режима *ECP* без RLE	0001 0000
3	Зарезервировано	0000 1000
2	Запрос идентификатора устройства с ответом в режиме:	
	полубайтный	0000 0100
	байтный	0000 0101
	ECP без RLE	0001 0100
	ECP с RLE	0011 0100
1	Зарезервировано	0000 0010
0	Запрос полубайтного режима	0000 0001
none	Запрос байтного режима	0000 0000

Последовательность согласования (рис. 1.8) состоит из следующих шагов:

1. Хост выводит байт расширяемости на линии данных.
2. Хост устанавливает высокий уровень сигнала SelectIn# и низкий — AutoFeed#, что означает начало последовательности согласования.
3. ПУ отвечает установкой низкого уровня сигнала Ack# и высокого — Error#, PaperEnd и Select. Устройство, «не понимающее» стандарта 1284, ответа не даст, и дальнейшие шаги не выполнятся.
4. Хост устанавливает низкий уровень сигнала Strobe# для записи байта расширяемости в ПУ.
5. Хост устанавливает высокий уровень сигналов Strobe# и AutoLF#.

6. ПУ отвечает установкой в низкий уровень сигналов PaperEnd и Error#, если ПУ имеет обратный канал передачи данных. Если запрошенный режим поддерживается устройством, на линии Select устанавливается высокий уровень, если не поддерживается — низкий.
7. ПУ устанавливает высокий уровень на линии Ack# для указания на завершение последовательности согласования, после чего контроллер устанавливает требуемый режим работы.

Рис. 1.8. Последовательность согласования режимов IEEE 1284

1.2.4. Развитие стандарта IEEE 1284

Кроме основного стандарта IEEE 1284, который уже принят, в настоящее время в стадии проработки находятся новые стандарты, дополняющие его. К ним относятся:

- *IEEE P1284.1* «Standard for Information Technology for Transport Independent Printer/Scanner Interface (TIP/SI)». Этот стандарт разрабатывается для управления и обслуживания сканеров и принтеров на основе протокола NPAP (Network Printing Alliance Protocol).
- *IEEE P1284.2* «Standard for Test, Measurement and Conformance to IEEE Std. 1284» — стандарт для тестиро-

вания портов, кабелей и устройств на совместимость с IEEE 1284.

- *IEEE P1284.3* «Standard for Interface and Protocol Extensions to IEEE Std. 1284 Compliant Peripheral and Host Adapter Ports» — стандарт на драйверы и использование устройств прикладным программным обеспечением (ПО). Уже приняты спецификации BIOS для использования *EPP* драйверами DOS. Прорабатывается стандарт на разделяемое использование одного порта цепочкой устройств или группой устройств, подключаемых через мультиплексор.
- *IEEE P1284.4* «Standard for Data Delivery and Logical Channels for IEEE Std. 1284 Interfaces» направлен на реализацию пакетного протокола достоверной передачи данных через параллельный порт. Основой служит протокол MLC (Multiple Logical Channels) фирмы Hewlett-Packard, однако совместимость с ним в окончательной версии стандарта не гарантируется.

1.3. Применение параллельных интерфейсов и LPT-портов

Параллельные интерфейсы применяются в компьютерах разных семейств и классов, здесь мы ограничимся рассмотрением IBM PC-совместимых компьютеров.

1.3.1. Использование параллельных интерфейсов

Распространенным применением LPT-порта является **подключение принтера и плоттера**. Остановимся на аппаратных аспектах — режиме порта и кабеле подключения. Практически все принтеры могут работать с портом в режиме *SPP*, но применение расширенных режимов дает свои преимущества:

- Двунаправленный режим (*Bi-Di*) не повышает производительность, но служит для сообщения о состоянии и параметрах принтера.
- Скоростные режимы (*Fast Centronics*) повышают производительность принтера, но могут потребовать качественного кабеля (см. далее). От принтера не требуется каких-либо дополнительных «интеллектуальных» способностей.

- Режим *ECP* — потенциально самый эффективный, имеет системную поддержку во всех версиях Windows. На некоторых принтерах реализован не полностью (может отсутствовать аппаратная компрессия). *ECP* поддерживают принтеры HP DeskJet моделей 6xx, LaserJet 4 и далее, современные модели фирмы Lexmark. Требует применения кабеля, по частотным свойствам соответствующего IEEE 1284.

Простейший вариант *кабеля подключения принтера* — 18-проводный кабель с неперевитыми проводами. Он используется для работы в режиме *SPP*. При длине более 2 м желательно, чтобы хотя бы линии Strobe# и Busy были перевиты с отдельными общими проводами. Для скоростных режимов может оказаться непригодным, причем сбои могут происходить нерегулярно и лишь при определенных последовательностях передаваемых кодов. Встречаются кабели *Centronics*, у которых отсутствует связь контакта 17 разъема PC с контактом 36 разъема принтера. При попытке подключения таким кабелем принтера, работающего в стандарте 1284, появится сообщение о необходимости применения «двунаправленного кабеля». Принтер не может сообщить системе о поддержке расширенных режимов, на что рассчитывают драйверы принтера.

Неплохие электрические свойства имеют ленточные кабели, у которых сигнальные цепи (управляющих сигналов) чередуются с общими проводами. Но их применение в качестве внешнего интерфейса непрактично (нет второго защитного слоя изоляции, высокая уязвимость) и неэстетично (круглые кабели смотрятся лучше).

Идеальным вариантом являются кабели, в которых все сигнальные линии перевиты с общими проводами и заключены в общий экран — то, что требует IEEE 1248. Такие кабели гарантированно работают на скоростях до 2 Мбайт/с, их длина может достигать 10 м.

В табл. 1.12 приводится распайка *кабеля подключения принтера* с разъемом X1 типа *A* (DB25-P) со стороны PC и X2 типа *B* (*Centronics*-36) или типа *C* (миниатюрный) со стороны принтера. Использование общих проводов (GND) зависит от качества кабеля (см. выше). В простейшем случае

(18-проводный кабель) все сигналы GND объединяются в один провод. Качественные кабели требуют отдельного обратного провода для каждой сигнальной линии, однако в разъемах типа А и В для этого недостаточно контактов (см. табл. 1.12, где в скобках указаны номера контактов разъема PC типа А, которым соответствуют обратные провода). В разъеме типа С обратный провод (GND) имеется для каждой сигнальной цепи; сигнальным контактам 1–17 этого разъема соответствуют контакты GND 19–35.

Таблица 1.12. Кабель подключения принтера

X1, разъем PC типа A	Сигнал	X2, разъем PRN типа B	X2, разъем PRN типа C
1	Strobe#	1	15
2	Data0	2	6
3	Data1	3	7
4	Data2	4	8
5	Data3	5	9
6	Data4	6	10
7	Data5	7	11
8	Data6	8	12
9	Data7	9	13
10	Ack#	10	3
11	Busy	11	1
12	PaperEnd	12	5
13	Select	13	2
14	Auto LF#	14	17
15	Error#	32	4
16	Init#	31	14
17	Select In#	36	16
18	GND (1)	19	33
19	GND (2 3)	20 21	24 25
20	GND (4 5)	22 23	26 27
21	GND (6 7)	24 25	28 29
22	GND (8 9)	26 27	30 31
23	GND (11 15)	29	19 22
24	GND (10 12 13)	28	20 21 23
25	GND (14 16 17)	30	32 34 35

Ряд отечественных (и стран бывшего СЭВ) принтеров имеет интерфейс *ИРПР* (*IFSP* в документации на принтеры ROBOTRON). Он является близким родственником интерфейса *Centronics*, но со следующими отличиями:

- Линии данных инвертированы.
- Протокол квитирования несколько иной.
- Ко всем входным линиям (на принтере) подключены пары согласующих резисторов: 220 Ом к питанию +5 В и 330 Ом к общему проводу. Это позволяет использовать длинные кабели, но перегружает большинство интерфейсных адаптеров PC.
- Сигналы ошибки и конца бумаги отсутствуют.

Интерфейс ИРПР может быть программно реализован через обычный LPT-порт, но для устранения перегрузки выходных линий согласующие резисторы из принтера желательно удалить. Порт, перегруженный по выходу, может преподносить всякого рода сюрпризы (естественно, неприятные и трудно диагностируемые).

Для связи двух компьютеров по параллельному интерфейсу применяются различные кабели в зависимости от режимов используемых портов. Самый простой и медленный — полубайтный режим, работающий на *всех* портах. Для этого режима в кабеле достаточно иметь 10 сигнальных и один общий провод. Распайка разъемов кабеля приведена в табл. 1.13. Связь двух PC данным кабелем поддерживается стандартным ПО типа Interlnk из MS-DOS или Norton Commander.

Для машин PS/2 с *двунаправленным* портом фирма IBM выпускала переходное устройство в комплекте с программой Data Migration Facility. Переходник устанавливался на разъем LPT-порта PS/2, а к его разъему X2 типа *Centronics* присоединялся обычный принтерный кабель, подключаемый к LPT-порту любого PC. Так предлагалось решить проблему переноса файлов со старых компьютеров, оснащенных 5" дисководами, на компьютеры PS/2 с дисководами 3,5". Распайка такого переходника приведена в табл. 1.14. Как видно, данный переходник нельзя использовать при связи через Interlnk или Norton Commander. Если обе соединяемые машины имеют двуна-

правленные порты, переходник обеспечивает симметричную двунаправленную связь. По скорости обмена превосходит вышеописанное полубайтное соединение в 2 раза. Это соединение не соответствует двунаправленному режиму IEEE 1284.

Таблица 1.13. Кабель связи PC-PC (4-битный)

X1, разъем PC#1		X2, разъем PC#2	
Бит	**Контакт**	**Контакт**	**Бит**
DR.0	2	15	*SR.3*
DR.1	3	13	*SR.4*
DR.2	4	12	*SR.5*
DR.3	5	10	*SR.6*
DR.4	6	11	*SR.7*
SR.6	10	5	*DR.3*
SR.7	11	6	*DR.4*
SR.5	12	4	*DR.2*
SR.4	13	3	*DR.1*
SR.3	15	2	*DR.0*
GND	18–25	18–25	GND

Разъемы X1 и X2 — DB25-P (вилки).

Таблица 1.14. Переходник Data Migration для IBM PS/2

X1		X2	
Контакт	**Бит**	**Бит**	**Контакт**
1	*CR.0*	*SR.6*	10
2	*DR.0*	*DR.0*	2
3	*DR.1*	*DR.1*	3
4	*DR.2*	*DR.2*	4
5	*DR.3*	*DR.3*	5
6	*DR.4*	*DR.4*	6
7	*DR.5*	*DR.5*	7
8	*DR.6*	*DR.6*	8
9	*DR.7*	*DR.7*	9
10	*SR.6*	*CR.0*	1
12	*SR.5*	*CR.3*	36
17	*CR.3*	*SR.5*	12
18–25	GND	GND	19–30, 33

Разъемы X1 — DB25-P (вилка), X2 — *Centronics*-36 (розетка).

Высокоскоростная связь двух компьютеров может выполняться и в режиме *ECP* (режим *EPP* неудобен, поскольку требует синхронизации шинных циклов ввода/вывода двух компьютеров). В табл. 1.15 приведена распайка кабеля. В отличие от предыдущих таблиц, описывающих кабели для программно-управляемых режимов, в ней приведены имена сигналов, которые аппаратно генерируются адаптерами портов. Этот же кабель может использоваться и для связи в байтном режиме. Такая связь поддерживается Windows 95.

Таблица 1.15. Кабель связи PC-PC в режиме ECP и байтном режиме

Разъем X1		Разъем X2	
Контакт	**Имя в ECP**	**Имя в ECP**	**Контакт**
1	HostClk	PeriphClk	10
14	HostAck	PeriphAck	11
17	1284Active	PeriphRequest#	15
16	ReverseRequest#	AckReverse#	12
10	PeriphClk	HostClk	1
11	PeriphAck	HostAck	14
12	AckReverse#	ReverseRequest#	16
13	Xflag	–	–
15	PeriphRequest#	1284Active	17
2–9	Data [0:7]	Data [0:7]	2–9

Подключение сканера к LPT-порту эффективно, только если порт обеспечивает хотя бы двунаправленный режим (*Bi-Di*), поскольку основной поток — ввод. Лучше использовать порт *ECP*, если этот режим поддерживается сканером (или *EPP*, что маловероятно).

Подключение внешних накопителей (Iomega Zip Drive, CD-ROM и др.), *адаптеров ЛВС* и других симметричных устройств ввода/вывода имеет свою специфику. В режиме *SPP* наряду с замедлением работы устройства заметна принципиальная асимметрия этого режима: *чтение данных* происходит *в два раза медленнее, чем* (весьма небыстрая) *запись*. Применение *двунаправленного* режима (*Bi-Di* или *PS/2 Type 1*) устранит эту асимметрию — *скорости сравняются*. Только перейдя на *EPP*, можно получить *нормальную ско-*

рость работы. В режиме *EPP* подключение к LPT-порту почти не уступает по скорости подключению через ISA-контроллер. Это справедливо и при подключении устройств со стандартным интерфейсом шин к LPT-портам через преобразователи интерфейсов (например, LPT – IDE, LPT – SCSI, LPT – PCMCIA).

В табл. 1.16 описано назначение выводов разъема LPT-порта в различных режимах и их соответствие битам регистров стандартного порта.

Таблица 1.16. Назначение выводов разъема LPT-порта и бит регистров в режимах SPP, ECP и EPP

Контакт	I/O	Бит*	SPP	ECP	EPP
1	O/I	*CR.0*\	Strobe#	HostClk	Write#
2	O/I	*DR.0*	Data 0	Data 0	Data 0
3	O/I	*DR.1*	Data 1	Data 1	Data 1
4	O/I	*DR.2*	Data 2	Data 2	Data 2
5	O/I	*DR.3*	Data 3	Data 3	Data 3
6	O/I	*DR.4*	Data 4	Data 4	Data 4
7	O/I	*DR.5*	Data 5	Data 5	Data 5
8	O/I	*DR.6*	Data 6	Data 6	Data 6
9	O/I	*DR.7*	Data 7	Data 7	Data 7
10	I	*SR.6*	Ack#	PeriphClk	INTR#
11	I	*SR.7*\	Busy	PeriphAck	Wait#
12	I	*SR.5*	PaperEnd	AckReverse#	—**
13	I	*SR.4*	Select	Xflag	—**
14	O/I	*CR.1*\	Auto LF#	HostAck	DataStb#
15	I	*SR.3*	Error#	PeriphRequest#	—**
16	O/I	*CR.2*	Init#	ReverseRequest#	Reset#
17	O/I	*CR.3*\	Select In#	1284Active	AddrStb#

* Символом «\» отмечены инвертированные сигналы (1 в регистре соответствует низкому уровню линии).

** — означает «определяется пользователем».

1.3.2. Конфигурирование LPT-портов

Управление параллельным портом разделяется на два этапа — *предварительное конфигурирование* (Setup) аппаратных средств

порта и *текущее* (оперативное) *переключение* режимов работы прикладным или системным ПО. Оперативное переключение возможно только в пределах режимов, разрешенных при конфигурировании. Этим обеспечивается возможность согласования аппаратуры с ПО и блокирования ложных переключений, вызванных некорректными действиями программы.

Конфигурирование LPT-порта зависит от его исполнения. Порт, расположенный на плате расширения (мультикарте), устанавливаемой в слот ISA или ISA+VLB, конфигурируется джамперами на самой плате. Порт на системной плате конфигурируется через BIOS Setup.

Конфигурированию подлежат следующие параметры:

- *Базовый адрес* — 3BCh, 378h или 278h. При инициализации BIOS проверяет наличие портов по адресам именно в этом порядке и, соответственно, присваивает обнаруженным портам логические имена *LPT1*, *LPT2*, *LPT3*. Адрес 3BCh имеет адаптер порта, расположенный на плате MDA или HGC. Большинство портов по умолчанию конфигурируются на адрес 378h и могут переключаться на 278h.
- Используемая *линия запроса прерывания*: для *LPT* — IRQ7, для *LPT2* — IRQ5. Традиционно прерывания от принтера не используются, и этот дефицитный ресурс можно сэкономить. Однако при использовании скоростных режимов *ECP* (или *Fast Centronics*) работа через прерывания может заметно повысить производительность и снизить загрузку процессора.
- Использование *канала DMA* для режимов *ECP* и *Fast Centronics* — разрешение и номер канала DMA (по умолчанию — 3).

Режимы работы порта:

- *SPP* — порт работает только в стандартном однонаправленном программно-управляемом режиме.
- *PS/2*, он же *Bi-Directional* — отличается от *SPP* возможностью реверса канала (установкой *CR.5=1*).
- *Fast Centronics* — аппаратное формирование протокола *Centronics* с использованием FIFO-буфера и, возможно, DMA.

- *EPP* — в зависимости от использования регистров порт работает в режиме *SPP* или *EPP*.
- *ECP* — по умолчанию включается в режим *SPP* или *PS/2*, записью в *ECR* может переводиться в любой режим *ECP*, но перевод в *EPP* записью в *ECR* кода 100 не гарантируется.
- *ECP+EPP* — то же, что и *ECP*, но запись в *ECR* кода режима 100 переводит порт в *EPP*.

Выбор режима *EPP*, *ECP* или *Fast Centronics* сам по себе не приводит к повышению быстродействия обмена с подключенными ПУ, а только дает возможность драйверу и ПУ установить оптимальный режим в пределах их «разумения». Большинство современных драйверов и приложений пытаются использовать эффективные режимы, так что «подрезать им крылья» установкой простых режимов без веских оснований не стоит.

Принтеры и *сканеры* могут пожелать режима *ECP*. Windows (3.x, 95 и NT) имеет системные драйверы для этого режима. В среде DOS печать через *ECP* поддерживается только специальным загружаемым драйвером.

Сетевые адаптеры, внешние диски и *CD-ROM*, подключаемые к параллельному порту, могут использовать режим *EPP*. Для этого режима специальный драйвер пока еще не применяется; использование *EPP* включается в драйвер самого подключаемого устройства.

Большинство современных ПУ, подключаемых к LPT-порту, поддерживает стандарт 1284 и PnP. Для поддержки этих функций компьютером с аппаратной точки зрения достаточно иметь контроллер интерфейса, поддерживающий стандарт 1284. Если подключаемое устройство поддерживает PnP, оно по протоколу согласования режимов 1284 способно «договориться» с портом о возможных режимах обмена. Подключенное устройство должно сообщить операционной системе (ОС) все необходимые сведения о себе — идентификатор производителя, модель и набор поддерживаемых команд. Более подробная информация может содержать идентификатор класса, подробное описание и идентификатор устрой-

ства, с которым обеспечивается совместимость. В соответствии с принятой информацией ОС может предпринять действия по установке требуемого ПО для поддержки данного устройства.

1.3.3. Неисправности и тестирование параллельных портов

Тестирование параллельных портов разумно начинать с *проверки их наличия* в системе. Список адресов установленных портов появляется в таблице, выводимой BIOS на экран перед загрузкой ОС. Список можно посмотреть и с помощью тестовых программ или прямо в *BIOS Data Area* с помощью отладчика.

Если BIOS обнаруживает меньше портов, чем установлено физически, скорее всего, двум портам присвоен один адрес. При этом работоспособность ни одного из конфликтующих портов не гарантируется: они будут одновременно выводить сигналы, но при чтении состояния конфликт на шине скорее всего приведет к искажению данных. Программное тестирование порта без диагностической заглушки (Loop Back) не покажет ошибок, поскольку при этом читаются данные выходных регистров, а они у всех конфликтующих (по отдельности исправных) портов совпадут. Именно такое тестирование производит BIOS при проверке на наличие портов. Разбираться с такой ситуацией следует, последовательно устанавливая порты и наблюдая за адресами, появляющимися в списке.

Если физически установлен только один порт, а BIOS его не обнаруживает, то либо порт отключен при конфигурировании, либо он вышел из строя (скорее всего из-за нарушений правил подключения). Иногда вам везет, и неисправность устраняется при «передергивании» платы в слоте — там возникают проблемы с контактами.

Наблюдаются и такие «чудеса» — при «теплой» перезагрузке DOS после Windows 95 порт не виден (и приложения не могут печатать из MS-DOS). Однако после повторной перезагрузки DOS порт оказывается на месте. С этим явлением легче смириться, чем бороться.

Тестирование портов с помощью диагностических программ позволяет проверить выходные регистры, а при использовании специальных заглушек — и входные линии. Поскольку количество выходных линий порта (12) и входных (5) различно, то полная проверка порта с помощью пассивной заглушки принципиально невозможна. Разные программы тестирования требуют применения разных заглушек (рис. 1.9).

DB25P

Strobe#	1
Select	13
Data 0	2
Error#	15
Init#	16
Ack#	10
Select In#	17
Busy	11
Auto LF#	14
PaperEnd	12

Рис. 1.9. Схема заглушки для тестирования LPT-порта программой CheckIt

Большинство неприятностей при работе с LPT-портами доставляют *разъемы и кабели*. Для проверки порта, кабеля и принтера можно воспользоваться специальными тестами из популярных диагностических программ (CheckIt, PCCheck и т. п.), а можно и попытаться вывести на принтер какой-либо символьный файл.

- Если вывод файла с точки зрения DOS проходит (копирование файла на устройство с именем *LPTn* или *PRN* совершается быстро и успешно), а принтер (исправный) не напечатал ни одного символа — скорее всего, это обрыв (неконтакт в разъеме) цепи Strobe#.
- Если принтер находится в состоянии *On Line*, но появляется сообщение о его неготовности, причину следует искать в линии Busy.
- Если принтер, подключенный к порту, в стандартном режиме (*SPP*) печатает нормально, а при переходе в *ECP* начинаются сбои, следует проверить кабель — соответствует ли он требованиям IEEE 1284 (см. выше). Дешевые

кабели с неперевитыми проводами нормально работают на скоростях 50–100 Кбайт/с, но при скорости 1–2 Мбайт/с, обеспечиваемой ECP, имеют полное право не работать, особенно при длине более 2 м.

- Если при установке драйвера PnP-принтера появилось сообщение о необходимости применения «двунаправленного кабеля», проверьте наличие связи контакта 17 разъема DB-25 с контактом 36 разъема *Centronics.* Хотя эта связь изначально предусматривалась, в ряде кабелей она отсутствует.
- Если принтер искажает информацию при печати, возможен обрыв (или замыкание) линий данных. В этом случае удобно воспользоваться файлом, содержащим последовательность кодов всех печатных символов. Вот пример программы на языке Basic:

```
10 OPEN "bincod.chr" FOR OUTPUT AS #1
20 FOR J=2 TO 15
30 FOR I=0 TO 15
40    PRINT#1, CHR$(16*J+I);
50 NEXT I
60 PRINT#1
70 NEXT J
80 CLOSE #1
90 END
```

Файл *BINCOD.CHR*, созданный данной программой, представляет собой таблицу всех печатных символов (управляющие коды пропущены), расположенных по 16 символов в строке. Если файл печатается с повтором некоторых символов или их групп, по периодичности повтора можно легко вычислить оборванный провод данных интерфейса. Этот же файл удобно использовать для проверки аппаратной русификации принтера.

Аппаратные прерывания от LPT-порта используются не всегда. Даже DOS-программа фоновой печати *PRINT* работает с портом по опросу состояния, а ее обслуживающий процесс запускается по прерыванию от таймера. Поэтому неисправности, связанные с цепью прерывания от порта, проявляют-

ся не часто. Однако по-настоящему многозадачные ОС (например, NetWare) стараются работать с портом по прерываниям. Протестировать линию прерывания можно, только подключив к порту ПУ или заглушку. Если к порту с неисправным каналом прерывания подключить адаптер локальной сети, то он, возможно, будет работать, но с очень низкой скоростью: на любой запрос ответ будет приходить с задержкой в десятки секунд — принятый из адаптера пакет будет приниматься не по прерыванию (сразу по приходу), а по внешнему тайм-ауту.

1.3.4. Функции BIOS для LPT-порта

BIOS обеспечивает поддержку LPT-порта, необходимую для организации *вывода по интерфейсу Centronics.*

В процессе начального тестирования POST BIOS проверяет наличие параллельных портов по адресам 3BCh, 378h и 278h и помещает базовые адреса обнаруженных портов в ячейки *BIOS Data Area 0:0408h, 040Ah, 040Ch, 040Eh.* Эти ячейки хранят адреса портов *LPT1–LPT4*, нулевое значение адреса является признаком отсутствия порта с данным номером. В ячейки 0:0478, 0479, 047A, 047B заносятся константы, задающие тайм-аут для этих портов.

Поиск портов обычно ведется достаточно примитивно — по базовому адресу (в регистр данных предполагаемого порта) выводится тестовый байт (AAh или 55h), затем производится ввод по тому же адресу. Если считанный байт совпал с записанным, предполагается, что найден LPT-порт; его адрес помещается в ячейку *BIOS Data Area*. Базовые адреса портов могут быть впоследствии изменены программно. Адрес порта *LPT4* BIOS самостоятельно установить не может, поскольку в списке стандартных адресов поиска имеются только три указанных.

Обнаруженные порты *инициализируются* — записью в регистр управления формируется и снимается сигнал Init#, после чего записывается значение 0Ch, соответствующее исходному состоянию сигналов интерфейса. В некоторых случаях сигнал Init# активен с момента аппаратного сброса до инициализации порта во время загрузки ОС. Это можно за-

метить по поведению включенного принтера во время перезагрузки компьютера — у принтера надолго гаснет индикатор *On-Line*. Следствие этого явления — невозможность распечатки экранов (например, параметров BIOS Setup) по клавише *Print Screen* до загрузки ОС.

Программное прерывание BIOS *INT 17h* обеспечивает следующие функции поддержки LPT-порта:

- 00h — *вывод символа* из регистра *AL* по протоколу *Centronics* (без аппаратных прерываний). Данные помещаются в выходной регистр, и, дождавшись готовности принтера (снятия сигнала Busy), формируется строб.
- 01h — *инициализация* интерфейса и принтера (установка исходных уровней управляющих сигналов, формирование импульса Init#, запрет аппаратных прерываний и переключение на вывод двунаправленного интерфейса).
- 02h — *опрос состояния* принтера (чтение регистра состояния порта).

При вызове *INT 17h* номер функции задается в регистре *AH*, номер порта — в регистре *DX* (0 — *LPT1*, 1 — *LPT2*...). При возврате регистр *AH* содержит *код состояния* — биты регистра состояния *SR[7:3]* (биты 6 и 3 инвертированы) и флаг тайм-аута в бите 0. Флаг тайм-аута устанавливается при неудачной попытке вывода символа, если сигнал Busy не снимается в течение времени, определенного для данного порта в ячейках тайм-аута. В этом случае, согласно протоколу *Centronics*, строб данных не вырабатывается.

Перехват прерывания *INT 17h* является удобным способом внедрения собственных драйверов принтера. Потребность в них может возникать при подключении к порту принтера с интерфейсом *ИРПР* или необходимости перекодировки символов.

2. Последовательные интерфейсы

Последовательный интерфейс для передачи данных использует одну сигнальную линию, по которой информационные биты передаются друг за другом последовательно. Отсюда — название интерфейса и порта. Английские термины — *Serial Interface* и *Serial Port* (иногда их неправильно переводят как «серийные»). Последовательная передача позволяет сократить количество сигнальных линий и увеличить дальность связи. Характерной особенностью является применение не-ТТЛ сигналов. В ряде последовательных интерфейсов применяется гальваническая развязка внешних (обычно входных) сигналов от схемной земли устройства, что позволяет соединять устройства, находящиеся под разными потенциалами. Ниже будут рассмотрены интерфейсы RS-232C, RS-422A, RS-423A, RS-485, токовая петля, MIDI, а также COM-порт.

2.1. Способы последовательной передачи

Последовательная передача данных может осуществляться в асинхронном или синхронном режимах. При *асинхронной* передаче каждому байту предшествует *старт-бит*, сигнализирующий приемнику о начале посылки, за которым следуют *биты данных* и, возможно, *бит паритета* (четности). Завершает посылку *стоп-бит*, гарантирующий паузу между посылками (рис. 2.1). Старт-бит следующего байта посылается в любой момент после стоп-бита, то есть между передачами возможны паузы произвольной длительности. Старт-бит, имеющий всегда строго определенное значение (логический 0), обеспечивает простой механизм синхронизации приемника по сигналу от передатчика. Подразумевается, что приемник и передатчик работают на одной скорости обмена. Внутренний генератор синхронизации приемника использует счетчик-делитель опорной частоты, обнуляемый в момент приема начала старт-бита. Этот счетчик генерирует внутренние стробы, по которым приемник фиксирует последующие принимаемые

биты. В идеале стробы располагаются в середине битовых интервалов, что позволяет принимать данные и при незначительном рассогласовании скоростей приемника и передатчика. Очевидно, что при передаче 8 бит данных, одного контрольного и одного стоп-бита предельно допустимое рассогласование скоростей, при котором данные будут распознаны верно, не может превышать 5%. С учетом фазовых искажений и дискретности работы внутреннего счетчика синхронизации реально допустимо меньшее отклонение частот. Чем меньше коэффициент деления опорной частоты внутреннего генератора (чем выше частота передачи), тем больше погрешность привязки стробов к середине битового интервала, и требования к согласованности частот становятся более строгими. Чем выше частота передачи, тем больше влияние искажений фронтов на фазу принимаемого сигнала. Взаимодействие этих факторов приводит к повышению требований к согласованности частот приемника и передатчика с ростом частоты обмена.



Рис. 2.1. Формат асинхронной передачи

Формат асинхронной посылки позволяет выявлять возможные *ошибки передачи*:

- Если принят перепад, сигнализирующий о начале посылки, а по стробу старт-бита зафиксирован уровень логической единицы, старт-бит считается ложным и приемник снова переходит в состояние ожидания. Об этой ошибке приемник может и не сообщать.
- Если во время, отведенное под стоп-бит, обнаружен уровень логического нуля, фиксируется ошибка стоп-бита.
- Если применяется контроль четности, то после посылки бит данных передается *контрольный бит*. Этот бит до-

полняет количество единичных бит данных до четного или нечетного в зависимости от принятого соглашения. Прием байта с неверным значением контрольного бита приводит к фиксации ошибки.

Контроль формата позволяет обнаруживать обрыв линии: при этом принимаются логический нуль, который сначала трактуется как старт-бит, и нулевые биты данных, потом срабатывает контроль стоп-бита.

Для асинхронного режима принят ряд *стандартных скоростей обмена*: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19 200, 38 400, 57 600 и 115 200 бит/с. Иногда вместо единицы измерения «бит/с» используют «бод» (baud), но при рассмотрении двоичных передаваемых сигналов это некорректно. В бодах принято измерять частоту изменения состояния линии, а при недвоичном способе кодирования (широко применяемом в современных модемах) в канале связи скорости передачи бит (бит/с) и изменения сигнала (бод) могут отличаться в несколько раз (подробнее см. в приложении А).

Количество *бит данных* может составлять 5, 6, 7 или 8 (5- и 6-битные форматы распространены незначительно). Количество *стоп-бит* может быть 1, 1,5 или 2 («полтора бита» означает только длительность стопового интервала).

Асинхронный обмен в PC реализуется с помощью *COM-порта* с использованием протокола *RS-232C*.

Синхронный режим передачи предполагает постоянную активность канала связи. Посылка начинается с синхробайта, за которым сразу же следует поток информационных бит. Если у передатчика нет данных для передачи, он заполняет паузу непрерывной посылкой байтов синхронизации. Очевидно, что при передаче больших массивов данных накладные расходы на синхронизацию в данном режиме будут ниже, чем в асинхронном. Однако в синхронном режиме необходима внешняя синхронизация приемника с передатчиком, поскольку даже малое отклонение частот приведет к искажению принимаемых данных. Внешняя синхронизация возможна либо с помощью отдельной линии для передачи сигнала синхронизации, либо с использованием самосин-

хронизирующего кодирования данных, при котором на стороне приемника из принятого сигнала могут быть выделены импульсы синхронизации. В любом случае синхронный режим требует дорогих линий связи или оконечного оборудования. Для PC существуют специальные платы — адаптеры SDLC (дорогие), поддерживающие синхронный режим обмена. Они используются в основном для связи с большими машинами (mainframes) IBM и мало распространены. Из синхронных адаптеров в настоящее время применяются адаптеры интерфейса V.35.

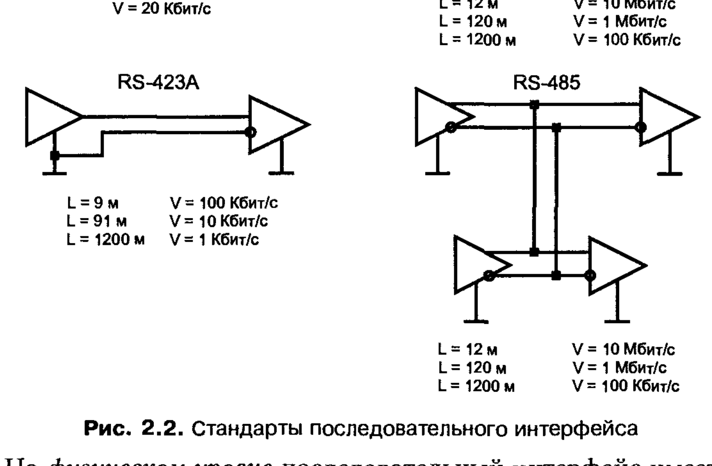


Рис. 2.2. Стандарты последовательного интерфейса

На *физическом уровне* последовательный интерфейс имеет различные реализации, различающиеся способом передачи электрических сигналов. Существует ряд родственных международных стандартов: *RS-232C*, *RS-423A*, *RS-422A* и *RS-485*. На рис. 2.2 приведены схемы соединения приемников и передатчиков, а также показаны ограничения на длину линии (L) и максимальную скорость передачи данных (V).

Несимметричные линии интерфейсов *RS-232C* и *RS-423A* имеют самую низкую защищенность от синфазной помехи, хотя дифференциальный вход приемника *RS-423A* несколь-

ко смягчает ситуацию. Лучшие параметры имеет двухточечный интерфейс *RS-422A* и его магистральный (шинный) аналог *RS-485*, работающие на симметричных линиях связи. В них для передачи каждого сигнала используются дифференциальные сигналы с отдельной (витой) парой проводов.

В перечисленных стандартах сигнал представляется ***потенциалом.*** Существуют последовательные интерфейсы, где информативен ток, протекающий по общей цепи передатчик-приемник — «токовая петля» и MIDI. Для связи на короткие расстояния приняты стандарты беспроводной инфракрасной связи. Наибольшее распространение в PC получил простейший из перечисленных — стандарт *RS-232C*, реализуемый COM-портами. В промышленной автоматике широко применяется *RS-485*, а также *RS-422A*, встречающийся и в некоторых принтерах. Существуют преобразователи сигналов для согласования этих родственных интерфейсов.

2.2. Интерфейс RS-232C

Интерфейс предназначен для подключения аппаратуры, передающей или принимающей данные (*ООД* — оконечное оборудование данных или *АПД* — аппаратура передачи данных; *DTE — Data Terminal Equipment*), к оконечной аппаратуре каналов данных (*АКД*; *DCE — Data Communication Equipment*). В роли АПД может выступать компьютер, принтер, плоттер и другое периферийное оборудование. В роли АКД обычно выступает модем. Конечной целью подключения является соединение двух устройств АПД. Полная схема соединения приведена на рис. 2.3. Интерфейс позволяет исключить канал удаленной связи вместе с парой устройств АПД, соединив устройства непосредственно с помощью нуль-модемного кабеля (рис. 2.4).

Стандарт описывает управляющие сигналы интерфейса, пересылку данных, электрический интерфейс и типы разъемов. В стандарте предусмотрены асинхронный и синхронный режимы обмена, но COM-порты поддерживают только

асинхронный режим. Функционально *RS-232C* эквивалентен стандарту МККТТ V.24/ V.28 и стыку С2, но они имеют различные названия сигналов.

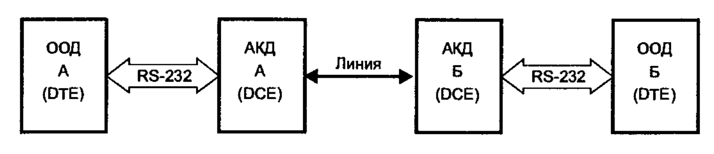


Рис. 2.3. Полная схема соединения по RS-232C

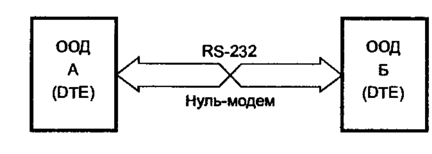


Рис. 2.4. Соединение по RS-232C нуль-модемным кабелем

2.2.1. Электрический интерфейс

Стандарт *RS-232C* использует несимметричные передатчики и приемники — сигнал передается относительно общего провода — схемной земли (симметричные дифференциальные сигналы используются в других интерфейсах — например, *RS-422*). Интерфейс *НЕ ОБЕСПЕЧИВАЕТ ГАЛЬВАНИЧЕСКОЙ РАЗВЯЗКИ* устройств. Логической единице соответствует напряжение на *входе приемника* в диапазоне −12...−3 В. Для линий управляющих сигналов это состояние называется *ON* («включено»), для линий последовательных данных — *MARK.* Логическому нулю соответствует диапазон +3...+12 В. Для линий управляющих сигналов состояние называется *OFF* («выключено»), а для линий последовательных данных — *SPACE.* Диапазон −3...+3 В — зона нечувствительности, обусловливающая гистерезис приемника: состояние линии будет считаться измененным только после пересечения порога (рис. 2.5). Уровни сигналов на выходах передатчиков должны быть в диапазонах −12...−5 В и +5...+12 В для представления единицы и нуля соответственно. Разность потенциалов между схемными землями (SG) соединяемых устройств должна быть менее 2 В, при более высокой разности потенциалов возможно неверное восприятие сигналов.

Интерфейс предполагает наличие *ЗАЩИТНОГО ЗАЗЕМЛЕНИЯ* для соединяемых устройств, если они оба питаются от сети переменного тока и имеют сетевые фильтры.

Подключение и отключение интерфейсных кабелей устройств с автономным питанием должно производиться *при отключенном питании.* Иначе разность невыровненных потенциалов устройств в момент коммутации может оказаться приложенной к выходным или входным (что опаснее) цепям интерфейса и вывести из строя микросхемы.

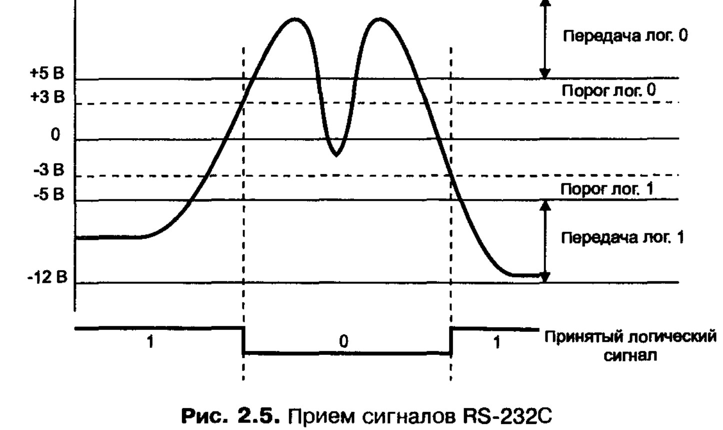


Рис. 2.5. Прием сигналов RS-232C

Для интерфейса *RS-232C* специально выпускаются буферные микросхемы приемников (с гистерезисом и передатчиком двуполярного сигнала). При несоблюдении правил заземления и коммутации они обычно являются первыми жертвами «пиротехнических» эффектов. Иногда их устанавливают в «кроватках», что облегчает замену. Цоколевка микросхем формирователей сигналов *RS-232C* приведена на рис. 2.6. Часто буферные схемы входят прямо в состав интерфейсных БИС. Это удешевляет изделие, экономит место на плате, но в случае аварии оборачивается крупными финансовыми потерями. Вывести из строя интерфейсные микросхемы замыканием сигнальных цепей маловероятно: ток короткого замыкания передатчиков обычно не превосходит 20 мА.

1A	1	1489	14	VCC	
1C	2		13	4A	
1Y	3		12	4C	
2A	4		11	4Y	
2C	5		10	3A	
2Y	6		9	3C	
GND	7		8	3Y	

а

VEE	1	1488	14	VDD
1A	2		13	4A
1Y	3		12	4B
2A	4		11	4Y
2B	5		10	3A
2Y	6		9	3B
GND	7		8	3Y

б

A	B*	Y
0	0	VDD
0	1	VDD
1	0	VDD
1	1	VEE

в

Рис. 2.6. Формирование сигналов RS-232C: *а* — приемник 1489 (А — вход RS-232, С — управление гистерезисом (ТТЛ), Y — выход ТТЛ); *б* — передатчик 1488 (А, В — входы ТТЛ, Y — выход RS-232, VDD = +12 В, VEE = −12 В); *в* — таблица состояния выходов передатчика (*1В — логическая 1)

Стандарт *RS-232C* регламентирует *типы применяемых разъемов.*

На аппаратуре *АПД* (в том числе на СОМ-портах) принято устанавливать *вилки* (male — «папа») *DB-25P* или более компактный вариант — *DB-9P*. Девятиштырьковые разъемы не имеют контактов для дополнительных сигналов, необходимых для синхронного режима (в большинстве 25-штырьковых разъемов эти контакты не используются).

На аппаратуре *АКД* (модемах) устанавливают *розетки* (female — «мама») *DB-25S* или *DB-9S.*

Это правило предполагает, что разъемы *АКД* могут подключаться к разъемам *АПД* непосредственно или через переходные «прямые» кабели с розеткой и вилкой, у которых контакты соединены «один в один». Переходные кабели могут являться и переходниками с 9- на 25-штырьковые разъемы (рис. 2.7).

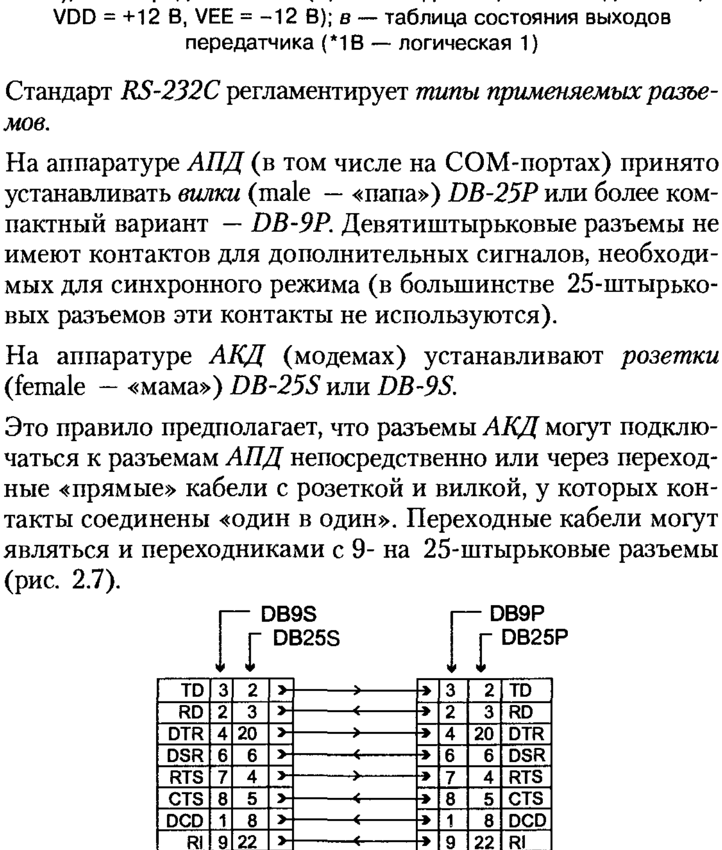


Рис. 2.7. Кабели подключения модемов

Если аппаратура *АПД* соединяется без модемов, то разъемы устройств (вилки) соединяются между собой *нуль-модемным кабелем* (Zero-modem или Z-modem), имеющим на обоих концах розетки, контакты которых соединяются перекрестно по одной из схем, приведенных на рис. 2.8.

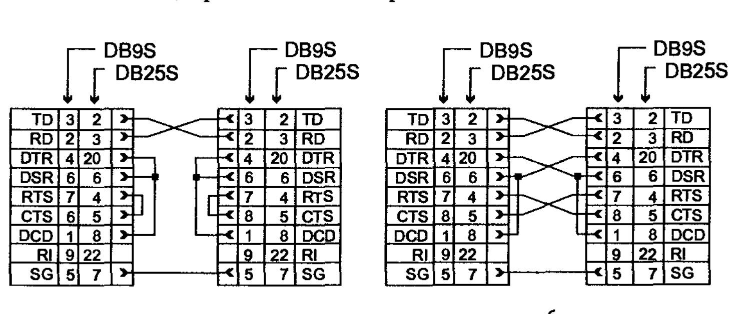


Рис. 2.8. Нуль-модемный кабель: *а* — минимальный, *б* — полный

Если на каком-либо устройстве *АПД* установлена розетка — это почти стопроцентный признак того, что к другому устройству оно должно подключаться прямым кабелем, аналогичным кабелю подключения модема. Розетка устанавливается обычно на тех устройствах, у которых удаленное подключение через модем не предусмотрено.

В табл. 2.1 приведено назначение контактов разъемов COM-портов (и любой другой аппаратуры *АПД*). Контакты разъема DB-25S определены стандартом EIA/TIA-232-E, разъем DB-9S описан стандартом EIA/ TIA-574. У модемов название цепей и контактов такое же, но роли сигналов (вход-выход) меняются на противоположные.

Подмножество сигналов *RS-232C*, относящихся к асинхронному режиму, рассмотрим с точки зрения COM-порта PC. Следует помнить, что активному состоянию сигнала («включено») и *логической единице* передаваемых данных соответствует *отрицательный потенциал* (ниже −3 В) сигнала интерфейса, а состоянию «выключено» и *логическому нулю* — *положительный* (выше +3 В). Назначение сигналов интерфейса приведено в табл. 2.2.

Таблица 2.1. Разъемы и сигналы интерфейса RS-232C

Обозначение цепи		Контакт разъема		Номер провода кабеля выносного разъема PC и контакта внутреннего разъема				Направление I/O
RS-232	Стык 2	DB-25S	DB-9S	1*	2*	3*	4*	
PG	101	7	5	(10)	(10)	(10)	1	—
SG	102	7	5	5	9	1	13	—
TD	103	2	3	3	5	3	3	O
RD	104	3	2	2	3	4	5	I
RTS	105	4	7	7	4	8	7	O
CTS	106	5	8	8	6	7	9	I
DSR	107	6	6	6	2	9	11	I
DTR	108/2	20	4	4	7	2	14	O
DCD	109	8	1	1	1	5	15	I
RI	125	22	9	9	8	6	18	I

1* — шлейф 8-битных мультикарт.

2* — шлейф 16-битных мультикарт и портов на системных платах.

3* — вариант шлейфа портов на системных платах.

4* — широкий шлейф к 25-контактному разъему.

Таблица 2.2. Назначение сигналов интерфейса RS-232C

Сигнал	Назначение
PG	*Protected Ground* — защитная земля, соединяется с корпусом устройства и экраном кабеля
SG	*Signal Ground* — сигнальная (схемная) земля, относительно которой действуют уровни сигналов
TD	*Transmit Data* — последовательные данные — выход передатчика
RD	*Receive Data* — последовательные данные — вход приемника
RTS	*Request To Send* — выход запроса передачи данных: состояние «включено» уведомляет модем о наличии у терминала данных для передачи. В полудуплексном режиме используется для управления направлением — состояние «включено» служит сигналом модему на переключение в режим передачи

Таблица 2.2 (продолжение)

Сигнал	Назначение
CTS	*Clear To Send* — вход разрешения терминалу передавать данные. Состояние «выключено» аппаратно запрещает передачу данных. Сигнал используется для аппаратного управления потоками данных
DSR	*Data Set Ready* — вход сигнала готовности от аппаратуры передачи данных (модем в рабочем режиме подключен к каналу и закончил действия по согласованию с аппаратурой на противоположном конце канала)
DTR	*Data Terminal Ready* — выход сигнала готовности терминала к обмену данными. Состояние «включено» поддерживает коммутируемый канал в состоянии соединения
DCD	*Data Carrier Detected* — вход сигнала обнаружения несущей удаленного модема
RI	*Ring Indicator* — вход индикатора вызова (звонка). В коммутируемом канале этим сигналом модем сигнализирует о принятии вызова

2.2.2. Управление потоком данных

Для управления потоком данных (Flow Control) могут использоваться два варианта протокола — аппаратный и программный. Иногда управление потоком путают с квитированием, но это разные методы достижения одной цели — согласования темпа передачи и приема. *Квитирование* (Handshaking) подразумевает посылку уведомления о получении элемента, в то время как *управление потоком* предполагает посылку уведомления о невозможности последующего приема данных.

Аппаратный протокол управления потоком RTS/CTS (Hardware Flow Control) использует сигнал CTS, который позволяет остановить передачу данных, если приемник не готов к их приему (рис. 2.9). Передатчик «выпускает» очередной байт только при включенной линии CTS. Байт, который уже начал передаваться, задержать сигналом CTS невозможно (это гарантирует целостность посылки). Аппаратный протокол обеспечивает самую быструю реакцию передатчика на состояние приемника. Микросхемы асинхронных приемопередатчиков имеют не менее двух регистров в приемной части —

сдвигающий, для приема очередной посылки, и хранящий, из которого считывается принятый байт. Это позволяет реализовать обмен по аппаратному протоколу без потери данных.



Рис. 2.9. Аппаратное управление потоком

Аппаратный протокол удобно использовать при подключении принтеров и плоттеров, если они его поддерживают (рис. 2.10). При непосредственном (без модемов) соединении двух компьютеров аппаратный протокол требует перекрестного соединения линий RTS — CTS.

Если аппаратный протокол не используется, у передающего терминала должно быть обеспечено состояние «включено» на линии CTS перемычкой RTS — CTS. В противном случае передатчик будет «молчать».

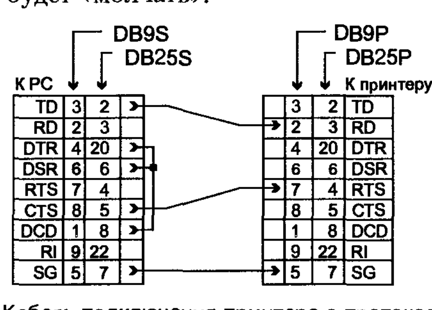


Рис. 2.10. Кабель подключения принтера с протоколом RTS-CTS

Программный протокол управления потоком XON/XOFF предполагает наличие двунаправленного канала передачи данных. Работает протокол следующим образом: если устройство, принимающее данные, обнаруживает причины, по которым не может их дальше принимать, оно по обратному последовательному каналу посылает байт-символ *XOFF* (13h). Противоположное устройство, приняв этот символ, приостанавливает передачу. Когда принимающее устройство снова становится готовым к приему данных, оно посылает символ

XON (11h), приняв который противоположное устройство возобновляет передачу. Время реакции передатчика на изменение состояния приемника по сравнению с аппаратным протоколом увеличивается по крайней мере на время передачи символа (*XON* или *XOFF*) плюс время реакции программы передатчика на прием символа (рис. 2.11). Из этого следует, что данные без потерь могут приниматься только приемником, имеющим дополнительный буфер принимаемых данных и сигнализирующим о неготовности заблаговременно (имея в буфере свободное место).

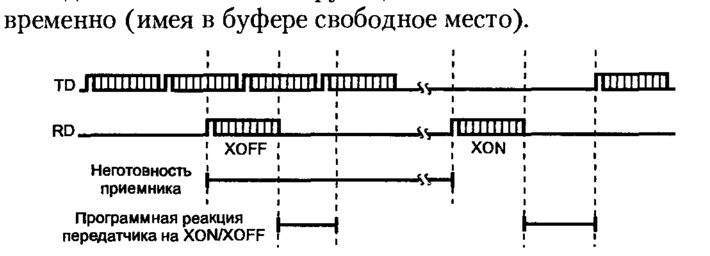


Рис. 2.11. Программное управление потоком XON/XOFF

Преимущество программного протокола заключается в отсутствии необходимости передачи управляющих сигналов интерфейса — минимальный кабель для двустороннего обмена может иметь только 3 провода (см. рис. 2.8а). Недостатком, кроме требования наличия буфера и большего времени реакции (снижающего общую производительность канала из-за ожидания сигнала *XON*), является сложность реализации полнодуплексного режима обмена. В этом случае из потока принимаемых данных должны выделяться (и обрабатываться) символы управления потоком, что ограничивает набор передаваемых символов. Минимальный вариант кабеля для подключения принтера (плоттера) с протоколом *XON/XOFF* приведен на рис. 2.12.

Кроме этих двух распространенных стандартных протоколов, поддерживаемых и ПУ, и ОС, существуют и другие. Некоторые плоттеры с последовательным интерфейсом используют программное управление, но посылают не стандартные символы *XON/XOFF*, а *слова* (ASCII-строки). Такой обмен на уровне системной поддержки протокола практически не поддерживается (эти плоттеры непосредственно

«разговаривают» с прикладной программой). Конечно, можно написать драйвер COM-порта (перехватчик *INT 14h*), но необходимость обработки в нем текстовых сообщений от устройства вывода обычно не вызывает восторга у системного программиста. Кабель для подключения совпадает с приведенным на рис. 2.12.

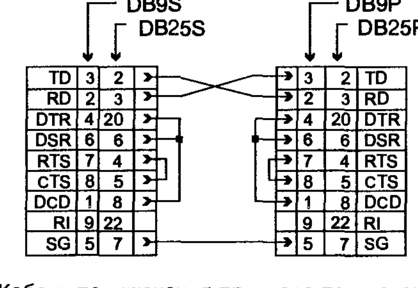


Рис. 2.12. Кабель подключения принтера по протоколу XON/XOFF

2.3. Интерфейс «токовая петля»

Распространенным вариантом последовательного интерфейса является токовая петля. В ней электрическим сигналом является не уровень напряжения относительно общего провода, а *ток* в двухпроводной линии, соединяющей приемник и передатчик. Логической единице (состоянию «включено») соответствует протекание тока 20 мА, а логическому нулю — отсутствие тока. Такое представление сигналов для описанного формата асинхронной посылки позволяет обнаружить обрыв линии — приемник заметит отсутствие стоп-бита (обрыв линии действует как постоянный логический нуль).

Токовая петля обычно предполагает *гальваническую развязку* входных цепей приемника от схемы устройства. При этом источником тока в петле является передатчик (этот вариант называют активным передатчиком). Возможно и питание от приемника (активный приемник), при этом выходной ключ передатчика может быть также гальванически развязан с остальной схемой передатчика. Существуют упрощенные варианты без гальванической развязки, но это уже вырожденный случай интерфейса.

Токовая петля с гальванической развязкой позволяет передавать сигналы на расстояния до нескольких километров. Расстояние определяется сопротивлением пары проводов и уровнем помех. Поскольку интерфейс требует пары проводов для каждого сигнала, обычно используют только два сигнала интерфейса. В случае двунаправленного обмена применяются только сигналы передаваемых и принимаемых данных, а для управления потоком используется программный метод *XON/XOFF*. Если двунаправленный обмен не требуется, используют одну линию данных, а для управления потоком обратная линия задействуется для сигнала CTS (аппаратный протокол) или встречной линии данных (программный протокол).

Преобразовать сигналы *RS-232C* в токовую петлю можно с помощью несложной схемы (рис. 2.13). Здесь принтер подключается по токовой петле к COM-порту с аппаратным управлением потоком. Для получения двуполярного сигнала, требуемого для входных сигналов COM-порта, применяется питание от интерфейса.

При надлежащем ПО одной токовой петлей можно обеспечить двунаправленную полудуплексную связь двух устройств. При этом каждый приемник «слышит» как сигналы передатчика на противоположной стороне канала, так и сигналы своего передатчика. Они расцениваются коммуникационными пакетами просто как эхо-сигнал. Для безошибочного приема передатчики должны работать поочередно.

Рис. 2.13. Подключение принтера с интерфейсом «токовая петля» к COM-порту

2.4. Интерфейс MIDI

Цифровой интерфейс музыкальных инструментов *MIDI* (Musical Instrument Digital Interface) является двунаправленным последовательным асинхронным интерфейсом с частотой передачи 31,25 Кбит/с. Этот интерфейс, разработанный в 1983 году, стал фактическим стандартом для сопряжения компьютеров, синтезаторов, записывающих и воспроизводящих устройств, микшеров, устройств специальных эффектов и другой электромузыкальной техники.

В интерфейсе применяется *токовая петля 10 мА* (возможно 5 мА) с гальванической развязкой входной цепи. Это исключает связь «схемных земель» соединяемых устройств через интерфейсный кабель, устраняя помехи, крайне нежелательные для звуковой техники. Снижению интерференционных помех служит и выбор частоты передачи, которая совпадает с одним из значений частот квантования, принятых в цифровой звукозаписи.

Асинхронная посылка содержит старт-бит, 8 бит информации и 1 стоп-бит, контроль четности отсутствует. Старший бит посылки является признаком «команда/данные». Его нулевое значение указывает на наличие семи бит данных в младших разрядах. При единичном значении биты [6:4] содержат *код команды*, а биты [3:0] — *номер канала*. Команды могут быть как адресованными конкретному каналу, так и широковещательными безадресными. К последней группе относятся команды старта, стопа и отметки времени, обеспечивающие синхронизацию устройств (система синхронизации *MIDI Sync* и *MTC — MIDI Time Code*).

Интерфейс определяет три типа портов: *MIDI-In*, *MIDI-Out* и *MIDI-Thru*.

Входной порт MIDI-In представляет собой вход интерфейса «токовая петля 10 мА», гальванически развязанного от приемника оптроном с быстродействием не хуже 2 мкс. Устройство отслеживает информационный поток на этом входе и реагирует на адресованные ему команды и данные.

Выходной порт MIDI-Out представляет собой выход источника тока 10 мА, гальванически связанного со схемой уст-

ройства. Ограничительные резисторы предохраняют выходные цепи от повреждения при замыкании на землю или источник 5 В. На выход подается информационный поток от данного устройства. В потоке может содержаться и транслированный входной поток.

Транзитный порт MIDI-Thru (не обязателен) служит для ретрансляции входного сигнала.

В качестве разъемов применяются 5-контактные разъемы DIN, распространенные в бытовой звуковой аппаратуре. На всех устройствах устанавливаются розетки, на кабелях — вилки. Все соединительные кабели MIDI унифицированы (рис. 2.14). Контакт 2 — экран кабеля — соединяется с общим проводом только на стороне передатчика (на разъемах *MIDI-Out* и *MIDI-Thru*).

Рис. 2.14. Соединительные кабели MIDI

В маркировке входов и выходов, указанной около разъемов, бывают разночтения. Одни производители пишут «In» или «Out» в соответствии с функцией разъема данного устройства (и это правильно), тогда любой кабель соединяет «In» и «Out». Другие считают, что подпись должна обозначать функцию подключаемого устройства. Тогда кабель будет соединять разъемы с обозначениями «In» — «In» и «Out» — «Out».

Интерфейс позволяет объединить группу до 16 устройств в локальную сеть. Топология должна подчиняться правилу: вход *MIDI-In* одного устройства должен подключаться к выходу *MIDI-Out* или *MIDI-Thru* другого. При планировании MIDI-сети необходимо руководствоваться информационными потоками и связью устройств. Управляющие устройства — клавиатуры, секвенсоры (в режиме воспроиз-

ведения), источники синхронизации — должны находиться перед управляемыми. Если устройства нуждаются в двунаправленном обмене, они соединяются в кольцо. Возможно применение специальных мультиплексоров, позволяющих логически коммутировать несколько входных потоков в один выходной. Вырожденным случаем кольца является двунаправленное соединение двух устройств. Несколько вариантов соединения приведено на рис. 2.15.

В PC MIDI-порт имеется на большинстве звуковых адаптеров, его сигналы выведены на неиспользуемые контакты (12 и 15) разъема игрового адаптера. Для подключения устройств MIDI требуется *переходной адаптер*, реализующий интерфейс «токовая петля». Переходной адаптер обычно встраивается в специальный кабель, схема которого приведена на рис. 2.16. Некоторые модели PC имеют встроенные адаптеры и стандартные 5-штырьковые разъемы MIDI.

Рис. 2.15. Варианты топологии сети MIDI: *а* — цепь, *б* — кольцо с мультиплексором

В PC для интерфейса MIDI применяются порты, совместимые с контроллером *MPU-401* (Roland) в режиме UART. В пространстве ввода/вывода *MPU-401* занимает два смежных адреса *MPU* (обычно 330h) и *MPU+1*:

- Порт *DATA* (адрес *MPU+0*) — запись и считывание байт, передаваемых и принимаемых по интерфейсу MIDI.

- Порт *STATUS/COMMAND* (адрес *MPU+1*) – чтение состояния / запись команд (запись – только для интеллектуального режима). В байте состояния определены следующие биты:

  Бит 7 – *DSR* (Data Set Ready) – готовность (*DSR*=0) принятых данных для чтения. Бит устанавливается в «1», когда все принятые байты считаны из регистра данных.

  Бит 6 – *DRR* (Data Read Ready) – готовность (*DRR*=0) UART к записи в регистр данных или команд. Условие готовности к записи не возникнет, если приемник имеет непрочитанный байт данных.

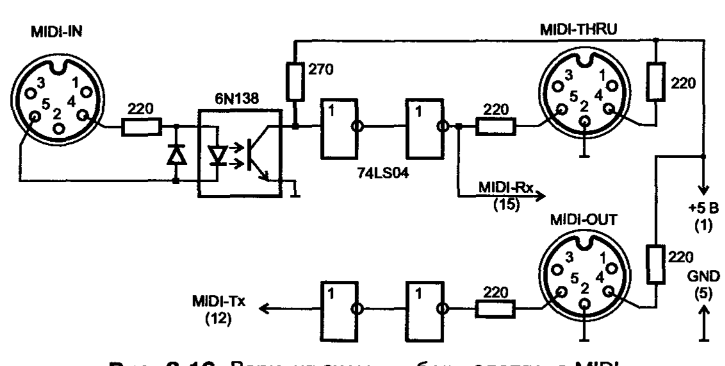


Рис. 2.16. Вариант схемы кабеля-адаптера MIDI

На некоторых системных платах применяются БИС контроллеров интерфейсов, в которых UART, используемая для COM-порта, конфигурированием через BIOS SETUP может быть переведена в режим MIDI-порта.

2.5. COM-порт

Последовательный интерфейс *COM-порт* (Communication Port – коммуникационный порт) появился в первых моделях IBM PC. Он был реализован на микросхеме асинхронного приемопередатчика Intel 8250. Порт имел поддержку BIOS (*INT 14h*), однако широко применялось (и применяется) взаимодействие с портом на уровне регистров. Поэтому во всех PC-совместимых компьютерах для последовательного интерфейса применяют микросхемы приемопередатчиков, совместимые с

i8250. В ряде отечественных PC-совместимых (почти) компьютеров для последовательного интерфейса применялась микросхема КР580ВВ51 – аналог i8251. Однако эта микросхема является универсальным синхронно-асинхронным приемопередатчиком (УСАПП или USART – Universal Asynchronous Receiver-Transmitter). Совместимости с PC на уровне регистров COM-порта такие компьютеры не имеют. Хорошо, если у соответствующих компьютеров имеется «честный» драйвер *BIOS INT 14h*, а не заглушка, возвращающая состояние модема «всегда готов» и ничего не делающая. Совместимость на уровне регистров COM-порта считается необходимой. Многие разработчики коммуникационных пакетов предлагают работу и через *BIOS INT 14h*, однако на высоких скоростях это неэффективно. Говоря о COM-порте PC, по умолчанию будем подразумевать совместимость регистровой модели с i8250 и реализацию асинхронного интерфейса *RS-232C*.

2.5.1. Использование COM-портов

COM-порты чаще всего применяют для **подключения манипуляторов** (мышь, трекбол). В этом случае порт используется в режиме последовательного ввода; питание производится от интерфейса. Мышь с последовательным интерфейсом – *Serial Mouse* –может подключаться к любому исправному порту. Для согласования разъемов порта и мыши возможно применение переходника DB-9S–DB-25P или DB-25S–DB-9P. Для мыши требуется прерывание, для порта *COM1* – IRQ4, для *COM2* – IRQ3. Жесткая привязка номера IRQ к номеру порта обусловлена свойствами драйверов. Каждое событие – перемещение мыши или нажатие-отпускание кнопки – кодируется двоичной посылкой по интерфейсу *RS-232C*. Применяется асинхронная передача; двуполярное питание обеспечивается от управляющих линий интерфейса (табл. 2.3).

Таблица 2.3. Разъемы Serial Mouse

Сигнал	Контакты	
	DB-9	DB-25
Data	2	3
GND	5	7
+V (питание)	4, 7	4, 20
–V (питание)	3	9

Две разновидности Serial Mouse — *MS–Mouse* и *PC–Mouse* (Mouse Systems Mouse) — требуют соответствующих драйверов, многие мыши имеют переключатель MS/PC. Мышь с «чужим» драйвером либо не отзывается, либо «скачет» загадочным образом. Эти разновидности используют различные форматы посылок: при одинаковой скорости 1200 бит/с, одном стоп-бите и отсутствии контроля четности Microsoft Mouse использует 7 бит данных, а PC-Mouse — 8 бит. Мышь посылает пакет при каждом изменении состояния — перемещении, нажатии или отпускании кнопки. Пакет, передаваемый MS-Mouse, состоит из трех байт (табл. 2.4). PC-Mouse передает 5 байт (табл. 2.5). Здесь *LB* (Left Buttom), *MB* (Middle Buttom) и *RB* (Right Buttom) означают состояние левой, средней и правой кнопок, *X[7:0]* и *Y[7:0]* — биты относительного перемещения мыши с момента предыдущей посылки по координатам X и Y. Положительным значениям соответствует перемещение по координате X вправо, а по координате Y вниз для MS-Mouse и вверх для PC-Mouse. Отсюда становятся понятными беспорядочные перемещения курсора на экране при несоответствии драйвера типу мыши.

Таблица 2.4. Формат пакета MS-Mouse

Биты	D6	D5	D4	D3	D2	D1	D0
1-й байт	1	*LB*	*RB*	*Y7*	*Y6*	*X7*	*X6*
2-й байт	0	*X5*	*X4*	*X3*	*X2*	*X1*	*X0*
3-й байт	0	*Y5*	*Y4*	*Y3*	*Y2*	*Y1*	*Y0*

Таблица 2.5. Формат пакета PC-Mouse (Mouse Systems Mouse)

Биты	D7	D6	D5	D4	D3	D2	D1	D0
1-й байт	1	0	0	0	0	*LB*	*MB*	*RB*
2-й байт	*X7*	*X6*	*X5*	*X4*	*X3*	*X2*	*X1*	*X0*
3-й байт	*Y7*	*Y6*	*Y5*	*Y4*	*Y3*	*Y2*	*Y1*	*Y0*
4-й байт	Совпадает со 2-м байтом							
5-й байт	Совпадает с 3-м байтом							

Для **подключения внешних модемов** используется полный (9-проводный) кабель *АПД–АКД*, схема которого приведена на рис. 2.7. Этот же кабель используется для согласования

разъемов (по количеству контактов); возможно применение переходников 9–25, предназначенных для мышей. Для работы коммуникационного ПО обычно требуется использование прерываний, но здесь есть свобода выбора номера (адреса) порта и линии прерывания. Если предполагается работа на скоростях 9600 бит/с и выше, то COM-порт должен быть реализован на микросхеме UART 16550A или совместимой. Возможности работы с использованием FIFO-буферов и обмена по каналам DMA зависят от коммуникационного ПО.

Для **связи двух компьютеров**, удаленных друг от друга на небольшое расстояние, используют и непосредственное соединение их COM-портов нуль-модемным кабелем (рис. 2.8). Использование программ типа Norton Commander или Interlnk MS-DOS позволяет обмениваться файлами со скоростью до 115,2 Кбит/с без применения аппаратных прерываний. Это же соединение может использоваться и сетевым пакетом Lantastic, предоставляющим более развитый сервис.

Подключение принтеров и плоттеров к COM-порту требует применения кабеля, соответствующего выбранному протоколу управления потоком: программному *XON/XOFF* или аппаратному RTS/CTS. Схемы кабелей приведены на рис. 2.10 и 2.12. Аппаратный протокол предпочтительнее. Прерывания при выводе средствами DOS (командами *COPY* или *PRINT*) не используются.

COM-порт используется для **подключения электронных ключей** (Security Devices), предназначенных для защиты от нелицензированного использования ПО. Эти устройства могут быть как «прозрачными», позволяя воспользоваться тем же портом для подключения периферии, так и полностью занимающими порт.

COM-порт при наличии соответствующей программной поддержки позволяет превратить PC в *терминал*, эмулируя систему команд распространенных специализированных терминалов (VT-52, VT-100 и т. д.). Простейший терминал получается, если замкнуть друг на друга функции BIOS обслуживания COM-порта (*INT 14h*), телетайпного вывода (*INT 10h*) и клавиатурного ввода (*INT 16h*). Однако такой терминал будет работать лишь на малых скоростях обмена (если,

конечно, его делать не на Pentium), поскольку функции BIOS хоть и универсальны, но не слишком быстры.

Интерфейс *RS-232C* широко распространен в различных ПУ и терминалах. COM-порт может использоваться и как двунаправленный интерфейс, у которого имеется 3 программно-управляемые выходные линии и 4 программно-читаемые входные линии с двуполярными сигналами. Их использование определяется разработчиком. Существует, например, схема однобитного широтно-импульсного преобразователя, позволяющего записывать звуковой сигнал на диск PC, используя входную линию COM-порта. Воспроизведение этой записи через обычный динамик PC позволяет передать речь. В настоящее время, когда звуковая карта стала почти обязательным устройством PC, это не впечатляет, но когда-то такое решение было интересным.

COM-порт используют для **беспроводных коммуникаций** с применением излучателей и приемников инфракрасного диапазона — *IR (Infra Red) Connection.* Этот интерфейс позволяет осуществлять связь между парой устройств, удаленных на расстояние, достигающее нескольких метров. Различают инфракрасные системы низкой (до 115,2 Кбит/с), средней (1,152 Мбит/с) и высокой (4 Мбит/с) скорости. Низкоскоростные системы служат для обмена короткими сообщениями, высокоскоростные — для обмена файлами между компьютерами, подключения к компьютерной сети, вывода на принтер, проекционный аппарат и т. п. Ожидаются более высокие скорости обмена, которые позволят передавать «живое видео». В 1993 году создана ассоциация разработчиков систем инфракрасной передачи данных *IrDA* (Infrared Data Association), призванная обеспечить совместимость оборудования от различных производителей. В настоящее время действует стандарт *IrDA 1.1.* Имеются собственные системы фирм Hewlett Packard — *HP-SIR* (Hewlett Packard Slow Infra Red) — и Sharp — *ASK* (Amplitude Shifted Keyed IR). Основные характеристики интерфейсов следующие:

- IrDA SIR (Slow Infra Red), HP-SIR — 9,6–115,2 Кбит/с;
- IrDA MIR (Middle Infra Red) — 1,2 Мбит/с;

- IrDA FIR (Fast Infra Red) — 4 Мбит/с;
- Sharp ASK — 9,6–57,6 Кбит/с.

На скоростях до 115 200 бит/с для инфракрасной связи используются UART, совместимые с 16450/16550. В современных системных платах на использование инфракрасной связи может конфигурироваться порт *COM2*. В этом случае на переднюю панель компьютера устанавливается внешний приемопередатчик — «инфракрасный глаз», который подключается к разъему *IR-Connector* системной платы.

На средних и высоких скоростях обмена применяются специализированные микросхемы, ориентированные на интенсивный программно-управляемый обмен или DMA с возможностью прямого управления шиной.

Инфракрасные излучатели не создают помех в радиочастотном диапазоне и обеспечивают конфиденциальность передачи. ИК-лучи не проходят через стены, поэтому зона приема ограничивается небольшим легко контролируемым пространством. Инфракрасная технология привлекательна для связи портативных компьютеров со стационарными компьютерами или док-станциями. Инфракрасный интерфейс имеют некоторые модели принтеров.

2.5.2. Микросхемы асинхронных приемопередатчиков

Преобразование параллельного кода в последовательный для передачи и обратное преобразование при приеме данных выполняют специализированные микросхемы UART (Universal Asynchronous Receiver-Transmitter — универсальный асинхронный приемопередатчик). Эти же микросхемы формируют и обрабатывают управляющие сигналы интерфейса. COM-порты IBM PC XT/AT базируются на микросхемах, совместимых на уровне регистров с UART i8250 — 8250/16450/16550A. Это семейство представляет собой усовершенствование начальной модели, направленное на повышение быстродействия, снижение потребляемой мощности и загрузки процессора при интенсивном обмене. Отметим следующее:

- 8250 имеет ошибки (появление ложных прерываний), учтенные в XT BIOS.

- 8250A — ошибки исправлены, в результате чего потеряна совместимость с BIOS. Работает в некоторых моделях AT, но непригоден для скорости 9600 бит/с.
- 8250B — исправлены ошибки 8250 и 8250A, восстановлена ошибка в прерываниях — возвращена совместимость с XT BIOS. Работает в AT под DOS (кроме скорости 9600 бит/с).

Микросхемы 8250x имеют невысокое быстродействие по обращениям со стороны системной шины. Они не допускают обращения к своим регистрам в смежных шинных циклах процессора — для корректной работы с ними требуется введение программных задержек между обращениями CPU.

В компьютерах класса AT применяют микросхемы UART следующих модификаций:

- 16450 — быстродействующая версия 8250 для AT. Ошибок 8250 и полной совместимости с XT BIOS не имеет. Требуется для работы OS/2 с COM-портами.
- 16550 — развитие 16450. Может использовать канал DMA для обмена данными. Имеет FIFO-буфер, но некорректность его работы не позволяет им воспользоваться.
- 16550A — имеет работающие 16-байтные FIFO-буферы приема и передачи и возможность использования DMA. Именно этот тип UART должен применяться в AT при интенсивных обменах на скоростях 9600 бит/с и выше. Совместимость с этой микросхемой обеспечивает большинство микросхем контроллеров портов ввода/вывода, входящих в современные чипсеты.

Микросхемы UART 16550A с программной точки зрения представляют собой набор регистров, доступ к которым определяется адресом (смещением адреса регистра относительно базового адреса порта) и значением бита ***DLAB*** (бита 7 регистра ***LCR***). В адресном пространстве микросхема занимает 8 смежных адресов. Список регистров UART 16550A и способы доступа к ним приведены в табл. 2.6. Микросхемы 8250 отличаются отсутствием регистра ***FCR*** и всех возможностей FIFO и DMA.

Таблица 2.6. Регистры UART 16550A

Доступ		Регистр		Чтение/запись R/W
Смещение	DLAB	Имя	Название	
0h	0	*THR*	*Transmit Holding Register*	WO
0h	0	*RBR*	*Receiver Buffer Register*	RO
0h	1	*DLL*	*Divisor Latch LSB*	R/W
1h	1	*DLM*	*Divisor Latch MSB*	R/W
1h	0	*IER*	*Interrupt Enable Register*	R/W
2h	x	*IIR*	*Interrupt Identification Register*	RO
2h	x	*FCR*	*FIFO Control Register*	WO
3h	x	*LCR*	*Line Control Register*	R/W
4h	x	*MCR*	*Modem Control Register*	R/W
5h	x	*LSR*	*Line Status Register*	R/W*
6h	x	*MSR*	*Modem Status Register*	R/W*
7h	x	*SCR*	*Scratch Pad Register*	R/W

* Некоторые биты допускают только чтеиие. Запись в регистр может привести к сбою протокола.

THR — промежуточный регистр данных передатчика (только для записи). Данные, записанные в регистр, будут пересланы в выходной сдвигающий регистр (когда он будет свободен), из которого поступят на выход при наличии разрешающего сигнала CTS. Бит 0 передается (и принимается) первым. При длине посылки менее 8 бит старшие биты игнорируются.

RBR — буферный регистр принимаемых данных (только для чтения). Данные, принятые входным сдвигающим регистром, помещаются в регистр *RBR*, откуда они могут быть считаны процессором. Если к моменту окончания приема очередного символа предыдущий не был считан из регистра, фиксируется ошибка переполнения. При длине посылки менее 8 бит старшие биты в регистре имеют нулевое значение.

DLL — регистр младшего байта делителя частоты.

DLM — регистр старшего байта делителя частоты. Делитель определяется по формуле D=115200/V, где V — скорость пе-

редачи, бит/с. Входная частота синхронизации 1,8432 МГц делится на заданный коэффициент, после чего получается 16-кратная частота передачи данных.

IER — регистр разрешения прерываний. Единичное значение бита разрешает прерывание от соответствующего источника.

Назначение бит регистра *IER*:

- Биты 7–4=0 — не используются.
- Бит 3 – *Mod IE* — по изменению состояния модема (любой из линий CTS, DSR, RI, DCD).
- Бит 2 – *RxL IE* — по обрыву/ошибке линии.
- Бит 1 – *TxD IE* — по завершении передачи.
- Бит 0 – *RxD IE* — по приему символа (в режиме FIFO — прерывание по тайм-ауту).

IIR — регистр идентификации (только для чтения) прерываний и признака режима FIFO. Для упрощения программного анализа UART выстраивает внутренние запросы прерывания по четырехуровневой системе приоритетов. Порядок приоритетов (по убыванию): состояние линии, прием символа, освобождение регистра передатчика, состояние модема. При возникновении условий прерывания UART указывает на источник с высшим приоритетом до тех пор, пока он не будет сброшен соответствующей операцией. Только после этого будет выставлен запрос с указанием следующего источника.

Назначение бит регистра *IIR*:

- Биты [7:6] – признак режима FIFO:

  11 – режим FIFO 16550A,

  10 – режим FIFO 16550,

  00 – обычный.
- Биты [5:4] — не используются.
- Бит 3 — прерывание по тайм-ауту (не в режиме FIFO).
- Биты [2:1] — причина прерывания с наивысшим приоритетом (в обычном, не FIFO-режиме):

  11 – ошибка/обрыв линии; сброс — чтением регистра состояния линии,

10 — принят символ; сброс — чтением данных,

01 — передан символ (регистр *THR* пуст); сброс — записью данных,

00 — изменение состояния модема; сброс — чтением регистра состояния модема.

- Бит 0 — признак необслуженного запроса прерывания:

  1 — нет запроса,

  0 — есть запрос.

Идентификация прерываний в режиме FIFO, *IIR* [3:1]:

- 011 — ошибка/обрыв линии; сброс — чтением регистра состояния линии.
- 010 — принят символ; сброс — чтением регистра данных приемника.
- 110 — индикатор тайм-аута (за 4-кратный интервал времени символа не передано и не принято ни одного символа, хотя в буфере имеется по крайней мере один). Сброс — чтением регистра данных приемника.
- 001 — регистр *THR* пуст; сброс — записью данных.
- 000 — изменение состояния модема (CTS, DSR, RI или DCD), сброс — чтением регистра *MSR*.

FCR — регистр управления FIFO (только для записи).

Назначение бит регистра *FCR*:

- Биты [7:6] — *ITL (Interrupt Trigger Level)* — уровень заполнения FIFO-буфера, при котором вырабатывается прерывание:

  00 — 1 байт (по умолчанию),

  01 — 4 байта,

  10 — 8 байт,

  11 — 14 байт.

- Биты [5:4] зарезервированы.
- Бит 3 — разрешение операций DMA.
- Бит 2 — *RESETTF (Reset Transmitter FIFO)* — сброс счетчика FIFO-передатчика (записью единицы; сдвигающий регистр не сбрасывается).

- Бит 1 — *RESETRF (Reset Receiver FIFO)* — сброс счетчика FIFO-приемника (записью единицы; сдвигающий регистр не сбрасывается).
- Бит 0 — *TRFIFOE (Transmit And Receive FIFO Enable)* — разрешение (единицей) режима FIFO для передатчика и приемника. При смене режима FIFO-буферы автоматически очищаются.

LCR — регистр управления линией (настройки параметров канала).

Назначения бит регистра *LCR*:

- Бит 7 — *DLAB (Divisor Latch Access Bit)* — управление доступом к делителю частоты.
- Бит 6 — *BRCON (Break Control)* — формирование обрыва линии (посылка нулей) при *BRCON*=1.
- Бит 5 — *STICPAR (Sticky Parity)* — принудительное формирование бита паритета:

  0 — контрольный бит генерируется в соответствии с паритетом выводимого символа,

  1 — постоянное значение контрольного бита: при *EVENPAR*=1 — нулевое, при *EVENPAR*=0 — единичное.
- Бит 4 — *EVENPAR (Even Parity* Select*)* — выбор типа контроля: 0 — нечетность, 1 — четность.
- Бит 3 — *PAREN (Parity Enable)* — разрешение контрольного бита:

  1 — контрольный бит (паритет или постоянный) разрешен,

  0 — запрещен.
- Бит 2 — *STOPB (Stop Bits)* — количество стоп-бит:

  0 — 1 стоп-бит,

  1 — 2 стоп-бита (для 5-битного кода стоп-бит будет иметь длину 1,5 бита).
- Биты [1:0] — *SERIALDB (Serial Data Bits)* — количество бит данных:

  00 — 5 бит, 01 — 6 бит, 10 — 7 бит, 11 — 8 бит.

MCR — регистр управления модемом.

Назначение бит регистра *MCR*:

- Биты [7:5]=0 — зарезервированы.
- Бит 4 — *LME (Loopback Mode Enable)* — разрешение режима диагностики:

  0 — нормальный режим,

  1 — режим диагностики (см. ниже).
- Бит 3 — *IE (Interrupt Enable)* — разрешение прерываний с помощью внешнего выхода *OUT2*; в режиме диагностики поступает на вход *MSR.7*:

  0 — прерывания запрещены,

  1 — разрешены.
- Бит 2 — *OUT1C (OUT1 Bit Control)* — управление выходным сигналом 1 (не используется); в режиме диагностики поступает на вход *MSR.6*.
- Бит 1 — *RTSC (Request To Send Control)* — управление выходом RTS; в режиме диагностики поступает на вход *MSR.4*:

  1 — активен (−V),

  0 — пассивен (+V).
- Бит 0 — *DTRC (Data Terminal Ready Control)* — управление выходом DTR; в режиме диагностики поступает на вход *MSR.5:*

  1 — активен (−V),

  0 — пассивен (+V).

LSR — регистр состояния линии (точнее, состояния приемопередатчика).

Назначение бит регистра *LSR*:

- Бит 7 — *FIFOE (FIFO Error Status)* — ошибка принятых данных в режиме FIFO (буфер содержит хотя бы один символ, принятый с ошибкой формата, паритета или обрывом). В не-FIFO-режиме всегда 0.
- Бит 6 — *TEMPT (Transmitter Empty Status)* — регистр передатчика пуст (нет данных для передачи ни в сдвиговом регистре, ни в буферных, *THR* или FIFO).

- Бит 5 — *THRE (Transmitter Holding Register Empty)* — регистр передатчика готов принять байт для передачи. В режиме FIFO указывает на отсутствие символов в FIFO-буфере передачи. Может являться источником прерывания.
- Бит 4 — *BD (Break Detected)* — индикатор обрыва линии (вход приемника находится в состоянии 0 не менее, чем время посылки символа).
- Бит 3 — *FE (Framing Error)* — ошибка кадра (неверный стоп-бит).
- Бит 2 — *PE (Parity Error)* — ошибка контрольного бита (паритета или фиксированного).
- Бит 1 — *OE (Overrun Error)* — переполнение (потеря символа). Если прием очередного символа начинается до того, как предыдущий был выгружен из сдвигающего регистра в буферный или FIFO, прежний символ в сдвигающем регистре теряется.
- Бит 0 — *DR (Receiver Data Ready)* — принятые данные готовы (в *DHR* или FIFO-буфере). Сброс — чтением приемника.

Индикаторы ошибок — биты [4:1] — сбрасываются после чтения регистра *LSR*. В режиме FIFO признаки ошибок хранятся в FIFO-буфере вместе с каждым символом. В регистре они устанавливаются (и вызывают прерывание) в тот момент, когда символ, принятый с ошибкой, находится на вершине FIFO (первый в очереди на считывание). В случае обрыва линии в FIFO заносится только один «обрывной» символ, и UART ждет восстановления и последующего старт-бита.

MSR — регистр состояния модема.

Назначение бит регистра *MSR*:

- Бит 7 — *DCD (Data Carrier Detect)* — состояние линии DCD.
- Бит 6 — *RI (Ring Indicator)* — состояние линии RI.
- Бит 5 — *DSR (Data Set Ready)* — состояние линии DSR.
- Бит 4 — *CTS (Clear To Send)* — состояние линии CTS.
- Бит 3 — *DDCD (Delta Data Carrier Detect)* — изменение состояния DCD.
- Бит 2 — *TERI (Trailing Edge Of Ring Indicator)* — спад огибающей RI (окончание звонка).

- Бит 1 — *DDSR (Delta Data Set Ready)* — изменение состояния DSR.
- Бит 0 — *DCTS (Delta Clear To Send)* — изменение состояния CTS.

Признаки изменения (биты [3:0]) сбрасываются по чтению регистра.

SCR — рабочий регистр (8 бит), на работу UART не влияет, предназначен для временного хранения данных (в 8250 отсутствует).

В *диагностическом режиме* (при *LME*=1) внутри UART организуется внутренняя заглушка:

- Выход передатчика переводится в состояние логической 1.
- Вход приемника отключается.
- Выход сдвигающего регистра передатчика логически соединяется со входом приемника.
- Входы DSR, CTS, RI и DCD отключаются от входных линий и внутренне управляются битами *DTRC*, *RTSC*, *OUT1C*, *IE*.
- Выходы управления модемом переводятся в пассивное состояние (логический ноль).

Переданные данные в последовательном виде немедленно принимаются, что позволяет проверять внутренний канал данных порта (включая сдвигающие регистры) и отработку прерываний, а также определять скорость работы UART.

2.5.3. Ресурсы и конфигурирование COM-портов

Компьютер может иметь до четырех последовательных портов *COM1–COM4* (для машин класса AT типично наличие двух портов). COM-порты имеют внешние разъемы-*вилки* *DB25P* или *DB9P*, выведенные на заднюю панель компьютера (назначение выводов приведено в табл. 2.1).

COM-порты реализуются на микросхемах *UART*, совместимых с семейством *i8250*. Они занимают в пространстве ввода/вывода по 8 смежных 8-битных регистров и могут располагаться по стандартным *базовым адресам.* Порты вырабатывают *аппаратные прерывания.* Возможность разделяемого использования одной линии запроса несколькими

портами (или ее разделения с другими устройствами) зависит от реализации аппаратного подключения и ПО. При использовании портов, установленных на шину ISA, разделяемые прерывания обычно не работают.

Управление последовательным портом разделяется на два этапа — предварительное конфигурирование (Setup) аппаратных средств порта и текущее (оперативное) переключение режимов работы прикладным или системным ПО. Конфигурирование COM-порта зависит от его исполнения. Порт на плате расширения конфигурируется джамперами на самой плате. Порт на системной плате конфигурируется через BIOS Setup.

Конфигурированию подлежат следующие параметры:

- *Базовый адрес*, который может иметь значения 3F8h, 2F8h, 3E8h (3E0h, 338h) или 2E8h (2E0h, 238h). При инициализации BIOS проверяет наличие портов по адресам именно в этом порядке и присваивает обнаруженным портам логические имена *COM1*, *COM2*, *COM3* и *COM4*. Для PS/2 стандартными для портов *COM3–COM8* являются адреса 3220h, 3228h, 4220h, 4228h, 5220h и 5228h соответственно.
- Используемая *линия запроса прерывания*: для *COM1* и *COM3* обычно используется IRQ4 или IRQ11, для *COM2* и *COM4* — IRQ3 или IRQ10. В принципе номер прерывания можно назначать в произвольных сочетаниях с базовым адресом (номером порта), но некоторые программы и драйверы (например, драйверы последовательной мыши) настроены на стандартные сочетания. Каждому порту, нуждающемуся в аппаратном прерывании, назначают отдельную линию, не совпадающую с линиями запроса прерываний других устройств. Прерывания необходимы для портов, к которым подключаются устройства ввода, UPS или модемы. При подключении принтера или плоттера прерываниями пользуются только многозадачные ОС (не всегда), и этот дефицитный ресурс PC можно сэкономить. Также прерываниями обычно не пользуются при связи двух компьютеров нуль-модемным кабелем.

- *Канал DMA* (для UART 16450/16550, расположенных на системной плате) — разрешение использования и номер канала DMA. Режим DMA при работе с COM-портами используют редко.

Режим работы порта по умолчанию (2400 бит/с, 7 бит данных, 1 стоп-бит и контроль четности), заданный при инициализации порта во время BIOS POST, может изменяться в любой момент при настройке коммуникационных программ или командой DOS *MODE COMx*: с указанием параметров.

2.5.4. Неисправности и тестирование COM-портов

Неполадки с COM-портами случаются (выявляются) при установке новых портов или неудачном подключении внешних устройств.

Проверка конфигурирования

Тестирование последовательных портов (как и параллельных) начинают с проверки их опознавания системой. Список *адресов* установленных портов обычно появляется в таблице, выводимой BIOS перед загрузкой ОС. Список можно посмотреть с помощью тестовых программ или прямо в BIOS Data AREA с помощью отладчика.

Если BIOS обнаруживает меньше портов, чем установлено физически, вероятно, двум портам присвоен один адрес или установлен нестандартный адрес какого-либо порта. Проблемы могут возникать с адресами портов *COM3* и *COM4*: не все версии BIOS будут искать порты по альтернативным адресам 3E0h, 338h, 2E0h и 238h; иногда не производится поиск по адресам 3E8h и 2E8h. Нумерация найденных портов, отображаемая в заставке, может вводить в заблуждение: если установлены два порта с адресами 3F8h и 3E8h, в заставке они могут называться *COM1* и *COM2*, и по этим именам на них можно ссылаться. Однако те же порты в заставке могут называться и *COM1* и *COM3* (поскольку 3E8h является штатным адресом для *COM3*), но попытка сослаться на порт *COM3* будет неудачной, поскольку в данном случае адрес 3E8h будет находиться в ячейке 0:402h *BIOS Data Area*, соответствующей порту *COM2*, а в ячейке порта *COM3* (0:404h) будет нуль — признак отсутствия такового порта.

«Объяснить» системе, где какой порт, можно вручную любым отладчиком, занеся правильные значения базовых адресов в ячейки *BIOS Data Area* (это придется делать каждый раз после перезагрузки ОС перед использованием «потерянного» порта). Существуют тестовые утилиты, позволяющие находить порты (например, Port Finder).

Если двум портам назначен один и тот же адрес, тестовая программа обнаружит ошибки порта только с использованием внешней заглушки (*External LoopBack*). Программное тестирование порта без заглушки не покажет ошибок, поскольку при этом включается диагностический режим (см. описание UART) и конфликтующие (по отдельности исправные) порты будут работать параллельно, обеспечивая совпадение считываемой информации. В реальной жизни нормальная работа конфликтующих портов невозможна. Разбираться с конфликтом адресов удобно, последовательно устанавливая порты и наблюдая за адресами, появляющимися в списке.

Если физически установлен только один порт и его не обнаруживает BIOS, причины те же, что с LPT-портом: либо он отключен при конфигурировании, либо вышел из строя. Неисправность может устраниться при вынимании/вставке платы адаптера в слот системной шины.

При работе с COM-портом задействуются соответствующие *аппаратные прерывания* — их используют при подключении модема, мыши и других устройств ввода. Неработоспособность этих устройств может быть вызвана некорректной настройкой запроса прерывания. Здесь возможны как конфликты с другими устройствами, так и несоответствие номера прерывания адресу порта.

Функциональное тестироввние

В первом приближении COM-порт можно проверить диагностической программой (CheckIt) без использования заглушек. Этот режим тестирования проверяет микросхему UART (внутренний диагностический режим) и вырабатывание прерываний, но не входные и выходные буферные микросхемы, которые являются более частыми источниками неприятностей. Если тест не проходит, причину следует искать

или в конфликте адресов/прерываний, или в самой микросхеме UART.

Для более достоверного тестирования рекомендуется использовать *внешнюю заглушку*, подключаемую к разъему COM-порта (рис. 2.17). В отличие от LPT-порта у COM-порта количество входных сигналов превышает количество выходных, что позволяет выполнить полную проверку всех цепей. Заглушка соединяет выход приемника со входом передатчика. Обязательная для всех схем заглушек перемычка RTS-CTS позволяет работать передатчику — без нее символы не смогут передаваться. Выходной сигнал DTR обычно используют для проверки входных линий DSR, DCD и RI.



Рис. 2.17. Заглушка для проверки COM-портов (LoopBack)

Если тест с внешней заглушкой не проходит, причину следует искать во внешних буферах, их питании или в шлейфах подключения внешних разъемов. Здесь может помочь осциллограф или вольтметр. Последовательность проверки может быть следующей:

1. Проверить наличие двуполярного питания выходных схем передатчиков (этот шаг логически первый, но поскольку он технически самый сложный, его можно отложить на крайний случай, когда появится желание заменить буферные микросхемы).

2. Проверить напряжение на выходах TD, RTS и DTR: после аппаратного сброса на выходе TD должен быть отрицательный потенциал около −12 В (по крайней мере ниже −5 В), а на выходах RTS и DTR — такой же положительный. Если этих потенциалов нет, возможна ошибка

подключения разъема к плате через шлейф. Распространенные варианты:

- шлейф не подключен;
- шлейф подключен неправильно (разъем перевернут или вставлен со смещением);
- раскладка шлейфа не соответствует разъему платы.

Первые два варианта проверяются при внимательном осмотре, третий же может потребовать некоторых усилий. В табл. 2.1 приведены три варианта раскладки 10-проводного шлейфа разъема COM-порта, известных автору; для COM-портов на системных платах возможно существование и других. Теоретически шлейф должен поставляться в соответствии с разъемом платы, на которой расположен порт.

Если дело в ошибочной раскладке, то эти три выходных сигнала удастся обнаружить на других контактах разъемов (на входных контактах потенциал совсем небольшой). Если эти сигналы обнаружить не удалось, очевидно, вышли из строя буферные формирователи.

3. Соединив контакты линий RTS и CTS (или установив заглушку), следует попытаться вывести небольшой файл на COM-порт (например, командой *COPY C:\AUTOEXEC.BAT COM1:*). С исправным портом эта команда успешно выполнится за несколько секунд с сообщением об успешном копировании. При этом потенциалы на выходах RTS и DTR должны измениться на отрицательные, а на выходе TD должна появиться пачка двуполярных импульсов с амплитудой более 5 В. Если потенциалы RTS и DTR не изменились, ошибка в буферных формирователях. Если на выходе RTS (и входе CTS) появился отрицательный потенциал, а команда *COPY* завершается с ошибкой, скорее всего, вышел из строя приемник линии CTS (или опять-таки ошибка в шлейфе). Если команда *COPY* успешно проходит, а изменения на выходе TD не обнаруживаются (их можно увидеть стрелочным вольтметром, но оценить амплитуду импульсов не удастся), виноват буферный передатчик сигнала *TD*.

Замена микросхем приемников и передатчиков существенно облегчается, если они установлены в «кроватки». Перед заменой следует с помощью осциллографа или вольтметра удостовериться в неисправности конкретной микросхемы. Расположение выводов микросхем приведено на рис. 2.6.

Если буферные элементы включены в состав интерфейсной БИС (что теперь весьма распространено), то такой порт ремонту не подлежит (по крайней мере в обычных условиях). Неисправный COM-порт, установленный на системной плате, можно попытаться отключить в BIOS SETUP, но порт мог сгореть и вместе со схемой своего отключения — тогда он останется «живым мертвецом» в карте портов ввода/вывода и прерываний. Иногда он полностью выводит из строя системную плату.

Источниками ошибок могут являться *разъемы* и *кабели*. В разъемах встречаются плохие контакты, а кабели кроме возможных обрывов могут иметь плохие частотные характеристики. Частотные свойства кабелей обычно сказываются при большой длине (десятки метров) на высоких скоростях обмена (56 или 115 Кбит/с). При необходимости использования длинных кабелей на высоких скоростях сигнальные провода данных должны быть перевиты с отдельными проводами «схемной земли».

Питание от интерфейса, или Почему может не работать мышь

При подключении к COM-порту устройств с небольшим энергопотреблением возникает соблазн использования питания от выходных линий интерфейса. Если линии управления DTR и RTS не используются по прямому назначению, их можно задействовать как питающие с напряжением около 12 В. Ток короткого замыкания на «схемную землю» ограничен буферной микросхемой передатчика на уровне 20 мА. При инициализации порта эти линии переходят в состояние «выключено», то есть вырабатывают *положительное* напряжение. Линия TD в покое находится в состоянии логической единицы, так что на выходе вырабатывается *отрицательное* напряжение. Потенциалами линий можно управлять через регистры COM-порта (выход TD вырабатывает положительное напряжение, если установить бит *BRCON*).

Двуполярным питанием от линий интерфейса (+V от DTR и RTS, −V от TD) пользуются все манипуляторы, подключаемые к COM-портам. Зная это, в случае неработоспособности мыши с данным портом следует проверить напряжения на соответствующих контактах разъема. Бывает, что с конкретным портом не работает только конкретная мышь (модель или экземпляр), хотя другие мыши с этим портом и эти же мыши с другими портами работают нормально. Здесь дело может быть в уровнях напряжений. Стандарт требует от порта выходного напряжения не менее 5 В (абсолютного значения), и, если данный порт обеспечивает только этот минимум, некоторым мышам не хватит мощности для питания светодиодов (главных потребителей энергии).

Порт получает двуполярное питание через системную плату от блока питания компьютера. Отсутствие на выходе блока питания напряжения +12 В обычно обнаружится по неработоспособности дисков. Отсутствие напряжения −12 В «заметят» только устройства, подключенные к COM-портам. Блок питания теоретически контролирует наличие этих напряжений на своем выходе (сообщая о неполадках сигналом Power Good, вызывающим аппаратный сброс). Встречаются упрощенные схемы блоков питания, у которых контролируются не все напряжения. Кроме того, возможны плохие контакты в разъеме подключения питания к системной плате.

2.5.5. Функции BIOS для COM-портов

В процессе начального тестирования POST BIOS проверяет наличие последовательных портов (регистров UART 8250 или совместимых) по стандартным адресам и помещает базовые адреса обнаруженных портов в ячейки *BIOS Data Area* 0:0400, 0402, 0404, 0406. Эти ячейки хранят адреса портов с логическими именами *COM1–COM4*. Нулевое значение адреса является признаком отсутствия порта с данным номером. В ячейки 0:047C, 047D, 047E, 047F заносятся константы, задающие тайм-аут для портов.

Обнаруженные порты *инициализируются* на скорость обмена 2400 бит/с, 7 бит данных с контролем на четность

(even), 1 стоп-бит. Управляющие сигналы интерфейса DTR и RTS переводятся в исходное состояние («выключено» — положительное напряжение).

Порты поддерживаются *сервисом BIOS INT 14h*, который обеспечивает следующие функции:

- 00h — *инициализация* (установка скорости обмена и формата посылок, заданных регистром *AL*; запрет источников прерываний). На сигналы DTR и RTS влияния не оказывает (после аппаратного сброса они пассивны).
- 01h — *вывод символа* из регистра *AL* (без аппаратных прерываний). Активируются сигналы DTR и RTS, и после освобождения регистра *THR* в него помещается выводимый символ. Если за заданное время регистр не освобождается, фиксируется ошибка тайм-аута и функция завершается.
- 02h — *ввод символа* (без аппаратных прерываний). Активируется только сигнал DTR (RTS переходит в пассивное состояние), и ожидается готовность принятых данных, принятый символ помещается в регистр *AL*. Если за заданное время данные не получены, функция завершается с ошибкой тайм-аута.
- 03h — *опрос состояния* модема и линии (чтение регистров *MSR* и *LSR*). Эту гарантированно быструю функцию обычно вызывают перед функциями ввода/вывода во избежание риска ожидания тайм-аута.

При вызове *INT 14h* номер функции задается в регистре *AH*, номер порта (0–3) — в регистре *DX* (0 — *COM1*, 1 — *COM2...*). При возврате из функций 0, 1 и 3 регистр *AH* содержит *байт состояния линии* (регистр *LSR*), *AL* — *байт состояния модема* (*MSR*). При возврате из функции 2 нулевое значение бита 7 регистра *AH* указывает на наличие принятого символа в регистре *AL*; ненулевое значение бита 7 — на ошибку приема, которую можно уточнить функцией 3.

Формат байта состояния линии (регистр *AH*):

- Бит 7 — ошибка тайм-аута (после вызова функции 2 — признак любой ошибки).

- Бит 6 — регистр сдвига передатчика пуст (пауза передачи).
- Бит 5 — промежуточный регистр передатчика пуст (готов принять символ для передачи).
- Бит 4 — обнаружен обрыв линии.
- Бит 3 — ошибка кадра (отсутствие стоп-бита).
- Бит 2 — ошибка паритета принятого символа.
- Бит 1 — переполнение (потеря символа).
- Бит 0 — регистр данных содержит принятый символ.

Формат байта состояния модема (регистр *AL*):

- Бит 7 — состояние линии DCD.
- Бит 6 — состояние линии RI.
- Бит 5 — состояние линии DSR.
- Бит 4 — состояние линии CTS.
- Бит 3 — изменение состояния DCD.
- Бит 2 — изменение огибающей RI.
- Бит 1 — изменение состояния DSR.
- Бит 0 — изменение состояния CTS.

Формат регистра *AL* при инициализации:

- Биты [7:5] — скорость обмена:
  000–110; 010–300; 100–1200; 110–4800,
  001–150; 011–600; 101–2400; 111–9600 бит/с.
- Биты [4:3] — контроль паритета:
  01 — число единиц нечетное,
  11 — четное,
  00 и 10 — без контроля.
- Бит 2 — количество стоп-бит: 0 — 1 бит, 1 — 2 бита (на скорости 110 бит/с — 1,5 стоп-бита).
- Биты [1:0] — длина посылки: 00 — 5 бит, 01 — 6 бит, 10 — 7 бит, 11 — 8 бит.

2.5.6. COM-порт и PnP

Современные ПУ, подключаемые к COM-порту, могут поддерживать спецификацию PnP. Основная задача ОС заключается в идентификации подключенного устройства, для чего разработан несложный протокол, реализуемый на любых COM-портах чисто программным способом (рис. 2.18):

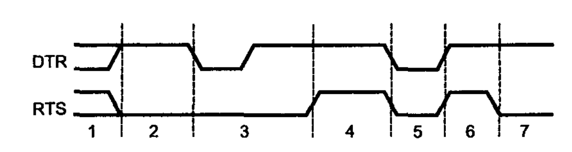


Рис. 2.18. Запрос идентификатора устройства PnP

1. Порт инициализируется с состоянием линий DTR=*ON*, RTS=*OFF*, *TXD=Mark* — состояние покоя (Idle).
2. Некоторое время (0,2 с) ожидается появление сигнала DSR, которое указало бы на наличие устройства, подключенного к порту. В простейшем случае устройство имеет на разъеме перемычку DTR-DSR, обеспечивающую указанный ответ. Если устройство обнаружено, выполняются манипуляции управляющими сигналами DTR и RTS для получения информации от устройства. Если ответ не получен, ОС, поддерживающая динамическое реконфигурирование, периодически опрашивает состояние порта для обнаружения новых устройств.
3. Порт программируется на режим 1200 бит/с, 7 бит данных, без паритета, 1 стоп-бит, и на 0,2 с снимается сигнал DTR. После этого устанавливается DTR=1, а еще через 0,2 с устанавливается и RTS=1.
4. В течение 0,2 с ожидается приход первого символа от устройства. По приходе символа начинается прием идентификатора (см. ниже). Если за это время символ не пришел (рис. 2.18), выполняется вторая попытка опроса (см. п. 5), несколько отличающаяся от первой.
5. На 0,2 с снимаются оба сигнала (DTR=0 и RTS=0), после чего они оба устанавливаются (DTR=1 и RTS=1).

6. В течение 0,2 с ожидается приход первого символа от устройства, по приходе символа начинается прием идентификатора (см. ниже). Если за это время символ не пришел, то в зависимости от состояния сигнала DSR переходят к проверке отключения *Verify Disconnect* (при DSR=0) или в дежурное состояние *Connect Idle* (при DSR=1).
7. В дежурном состоянии *Connect Idle* устанавливается DTR=1, RTS=0, порт программируется на режим 300 бит/с, 7 бит данных, без паритета, 1 стоп-бит. Если в этом состоянии обнаружится DSR=0, ОС следует уведомить об отключении устройства.

Посимвольный прием идентификатора устройства имеет ограничения по тайм-ауту в 0,2 с на символ, а также общее ограничение в 2,2 с, позволяющее принять строку длиной до 256 символов. Строка идентификатора PnP должна иметь маркеры начала (28h или 08h) и конца (29h или 09h), между которыми располагается тело идентификатора в стандартизованном формате. Перед маркером начала может находиться до 16 символов, не относящихся к идентификатору PnP. Если за первые 0,2 с ожидания символа (шаг 4 или 6) маркер начала не пришел, или же сработал тайм-аут, а маркер конца не получен, или же какой-либо символ принят с ошибкой, происходит переход в состояние *Connect Idle.* Если получена корректная строка идентификатора, она передается ОС.

Для *проверки отключения* (*Verify Disconnect*) устанавливается DTR=1, RTS=0 и через 5 с проверяется состояние сигнала DSR. При DSR=1 происходит переход в состояние *Connect Idle* (см. п. 7), при DSR=0 происходит переход в состояние *Disconnect Idle*, в котором система может периодически опрашивать сигнал DSR для обнаружения подключения устройства.

Описанный механизм разрабатывался фирмой Microsoft с учетом совместимости с не-PnP-устройствами — невозможность их вывода из строя и устойчивость системы к сообщениям, не являющимся PnP-идентификаторами. Например, обычная Microsoft Mouse при включении питания от интерфейса ответит ASCII-символом «M» (трехкнопочная — строкой «M3»).

3. Специальные интерфейсы

В этой главе рассматриваются следующие интерфейсы IBM PC-совместимых компьютеров: интерфейсы устройств ввода (клавиатуры и манипуляторов), аналоговый и дискретный интерфейсы игрового адаптера, аудиоинтерфейсы и интерфейсы мониторов.

3.1. Интерфейс клавиатуры

Для подключения клавиатуры предназначен последовательный синхронный интерфейс, состоящий из двух обязательных сигналов, KB-Data и KB-Clock. Необязательный сигнал KB-Reset сбрасывает клавиатуру низким уровнем сигнала. Интерфейс на системной плате XT реализован аппаратной логикой — регистром сдвига, параллельный выход которого подключается ко входам порта А системного интерфейса i8255. По приему байта от клавиатуры вырабатывается аппаратное прерывание IRQ2, обработчик которого может прочитать принятый байт из порта 60h. С помощью бит 7 и 6 порта 61h возможны программная блокировка и сброс клавиатуры соответственно. Сброс клавиатуры XT осуществляется обнулением линии KB-Clock.

Интерфейс клавиатуры AT построен на микроконтроллере i8042, обеспечивающем, в отличие от XT, двунаправленный интерфейс с клавиатурой. Передача информации к клавиатуре используется для управления индикаторами ее состояния и программирования параметров (автоповтор, набор скан-кодов).

Хотя электрический интерфейс клавиатур XT и AT совпадает (за исключением двунаправленного обмена в AT), логические форматы посылок существенно отличаются. POST способен производить диагностику клавиатуры. Подключение клавиатуры неподходящего типа или отсутствие клавиатуры он воспримет как ошибку. Если проверка клавиатуры

разрешена в BIOS Setup, то по этой ошибке POST будет дожидаться получения кода клавиши F1.

Вид разъемов клавиатур (со стороны задней панели) и назначение контактов приведены на рис. 3.1. Конструктивно возможны два варианта разъема — обычная 5-контактная розетка DIN или малогабаритная розетка mini-DIN (PS/2). На этот же разъем через плавкий предохранитель поступает напряжение питания клавиатуры +5 В. Электрически и логически интерфейс клавиатуры PS/2 повторяет AT, поэтому для согласования типа разъема применяют специальные переходники. Предпочтительнее использовать переходники, выполненные в виде мягкого кабеля с разъемами. Монолитный переходник, особенно с AT-клавиатуры на PS/2-разъем, хуже тем, что малейшее движение кабеля вызывает большой момент силы, выламывающий переходник из маленького гнезда PS/2.



Рис. 3.1. Разъемы подключения клавиатур XT, AT — *а* и PS/2 — *б* (вид со стороны контактов)

> Питание от разъема клавиатуры часто используется такими устройствами, как внешние накопители или адаптеры локальных сетей, подключаемыми к параллельному порту. Плавкий предохранитель, установленный на системной плате, может не выдержать броска тока, потребляемого этими устройствами. При этом откажется работать и клавиатура — ее индикаторы не мигнут при включении.

Программируемый микроконтроллер i8042 имеет встроенное ПО, которое хранится в масочном внутреннем ПЗУ и обеспечивает вырабатывание запроса прерывания по приему скан-кода от клавиатуры и отработку управляющих команд от ЦП. Кроме управления клавиатурой, через программно-управляемые и программно-читаемые линии внешних портов контроллера формируются сигналы управления вентилем *Gate A20*, аппаратного системного сброса, а также считываются сигналы от конфигурационных джамперов системной платы. Контроллер i8242B, кроме интерфейса кла-

виатуры, поддерживает аналогичный интерфейс дополнительного устройства, например PS/2-Mouse. Контроллер имеет два порта, доступ к которым осуществляется через команды контроллера.

Порт ввода, доступный по команде *C0h*, используется для чтения состояния джамперов и ключа:

- Бит 7 — если 0, то клавиатура заблокирована ключом (*KeyLock*).
- Бит 6 — исходный видеорежим: 0=CGA, 1=MDA.
- Бит 5 — системная перемычка: 0=замкнута.
- Бит 4 — системное ОЗУ: 0=512 Кбайт и более, 1=256 Кбайт.
- Бит 1 — вход данных дополнительного интерфейса.
- Бит 0 — вход данных интерфейса клавиатуры.

Порт вывода, доступный для записи и чтения по командам *D1h* и *D0h* соответственно, имеет следующее назначение бит:

- Бит 7 — последовательные данные клавиатуры.
- Бит 6 — синхронизация клавиатуры.
- Бит 5 — запрос прерывания от дополнительного интерфейса (IRQ12).
- Бит 4 — запрос прерывания от клавиатуры (IRQ1).
- Бит 3 — синхронизация дополнительного интерфейса.
- Бит 2 — последовательные данные дополнительного интерфейса.
- Бит 1 — вентиль линии адреса А20 (*Gate A20*).
- Бит 0 — альтернативный сброс процессора (без формирования общего сигнала сброса).

Контроллер расположен в пространстве ввода/вывода по адресам 60h и 64h, причем по чтению скан-кода клавиатуры из порта 60h сохраняется совместимость с XT. Назначение регистров контроллера клавиатуры приведено в табл. 3.1. Регистр данных контроллера в режиме записи используется для подачи команд, относящихся к клавиатуре и собственно контроллеру. Признаком готовности контроллера является нулевое значение бита 1 регистра состояния (порт 064h).

Таблица 3.1. Назначение регистров контроллера клавиатуры

Порт, R/W	Назначение
060 RW	*Порт данных 8042*
064 R	*Регистр состояния 8042*: Бит 7 — ошибка четности Бит 6 — тайм-аут приемника/общий тайм-аут* Бит 5 — тайм-аут передатчика/выходной буфер дополнительного интерфейса полон* Бит 4: 1 — клавиатура на замке Бит 3: 1 — команда, 0 — данные Бит 2: 0 — Power ON, 1 — Reset OK (состояние завершения системного сброса) Бит 1: 1 — входной буфер полон, 0 — готовность к приему команды/данных Бит 0: 1 — выходной буфер полон
064 W	*Регистр команд 8042*

* Второе назначение бита относится к контроллеру i8242B, имеющему дополнительный интерфейс для подключения PS/2-Mouse.

Общие команды контроллера i8042:

- *DF* — разрешить управление линией A20.
- *DD* — запретить управление линией A20.
- *D1* — запись в порт вывода i8042.
- *D0* — чтение порта вывода i8042.
- *C0* — чтение порта ввода i8042.
- *AE* — разрешение клавиатуры.
- *AD* — запрещение клавиатуры.
- *AC* — чтение ОЗУ контроллера.

- *AB* — тест синхронизации и данных.
- *AA* — внутренний тест контроллера.
- *60* — запись в контроллер.
- *20* — чтение контроллера.

Команды управления клавиатурой AT:

- *FF — Reset* — сброс и тест клавиатуры.
- *FE — Resend* — повтор последней передачи.
- *FB–FD — Set Key Types* — установка автоповтора отдельных клавиш.
- *F7–FA — Set All Keys* — установка операции автоповтора клавиатуры.
- *F6 — Set Default* — сброс в начальное состояние и разрешение сканирования.
- *F5 — Default Disable* — сброс в начальное состояние и запрет сканирования.
- *F4 — Enable KB* — клавиатура ответит байтом подтверждения *ACK* и продолжит сканирование.
- *F3 — Set Typematic Rate/Delay* — задание задержки и частоты повтора кодов. Последующий байт данных имеет следующий формат:

  Бит 7=0.

  Биты [6:5] — задержка, мс:

  00=250, 01=500, 10=750, 11=1000.

  Биты [4:0]=00–1Fh — скорость повторов, 1/с:

00h=30,0	05h=18,5	0Ah=12,0	12h=6,0
01h=26,7	06h=17,1	0Ch=10,0	14h=5,0
02h=24,0	07h=16,0	0Dh=9,2	17h=4,0
03h=21,8	08h=15,0	0Fh=8,0	1Ah=3,0
04h=20,0	09h=13,3	10h=7,5	1Fh=2,0

- *F2 — Read ID Code* — запрос на двухбайтный идентификатор.
- *EE — Echo* — эхо-диагностика: возвращает 0EEh.

- *ED — Set/Reset Mode Indicators* — управление индикаторами клавиатуры. Последующий байт данных:

  Бит 0: 1=включить Scroll Lock.

  Бит 1: 1=включить Num Lock.

  Бит 2: 1=включить Caps Lock.
- *AD* — запрет сканирования.
- *AE* — разрешение сканирования.

По нажатии клавиши ее скан-код попадает в регистр данных. По отпускании в регистр данных попадает сначала префикс отпускания, затем — скан код. В XT префикса нет, по отпускании передается скан-код с единицей в бите 7.

Из регистра данных контроллера кроме скан-кодов возможно получение ответов на команды, префикса кода отпускания клавиши (*F0*) или кодов особых случаев:

- *FA — ACK* — подтверждение приема команды.
- *AA — Test OK* — внутренний тест прошел.
- *FD — Diagnostic Failure* — ошибка внутреннего теста.
- *EE — Echo Response* — ответ на эхо-команду.
- *00, FF — Buffer Error* — переполнение буфера или неидентифицированная клавиша.

3.2. Интерфейсы манипуляторов

Устройство ввода *мышь* (mouse) передает в систему информацию о своем перемещении и нажатии-отпускании кнопок. Обычная конструкция имеет свободно вращающийся массивный обрезиненный шарик, передающий вращение на два координатных диска с фотоэлектрическими датчиками — две открытые оптопары (светодиод — фотодиод), в оптический канал которых входит вращающийся диск с прорезями.

По интерфейсу с компьютером различают три основных вида мышей: *Bus Mouse*, *Serial Mouse* и *PS/2-Mouse*. Ожидается появление мышей с интерфейсом USB.

Bus Mouse (шинная мышь) — вариант, применявшийся в первых мышах. Содержит только датчики и кнопки; обработка их сигналов производится на специализированной плате адаптера. Муль-

типортовые карты (COM-, LPT- и GAME-порты), на которых установлен и адаптер Bus Mouse, встречаются редко. Кабель 9-проводный, разъем специальный (см. рис. 3.2 и табл. 3.2), хотя на первый взгляд напоминает разъем PS/2-Mouse.

Таблица 3.2. Разъем Bus Mouse

Контакт	Сигналы	Контакт	Сигналы
1	Vcc (+5 В)	6	Lb — левая кнопка
2	Xa — датчик X	7	Mb — средняя кнопка
3	Xb — датчик X	8	Rb — правая кнопка
4	Ya — датчик Y	9	GND
5	Yb — датчик Y		

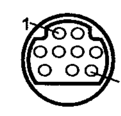


Рис. 3.2. Разъем Bus Mouse

Serial Mouse — мышь с последовательным интерфейсом, подключаемая через 25 или 9-штырьковый разъем COM-порта (табл. 2.3). Имеет встроенный микроконтроллер, который обрабатывает сигналы от координатных датчиков и кнопок. Каждое событие кодируется двоичной посылкой по интерфейсу RS-232C (см. 2.5.1).

PS/2-Mouse — мышь, появившаяся с компьютерами PS/2. Ее интерфейс и 6-штырьковый DIN мини-разъем аналогичны клавиатурному (рис. 3.3). Адаптер и разъем PS/2-Mouse устанавливается на современных системных платах brand-name. Контроллер такой мыши может входить в контроллер клавиатуры или занимать дополнительные адреса в пространстве ввода/вывода. Для PS/2-Mouse используется прерывание IRQ12.

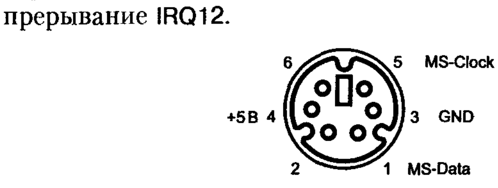


Рис. 3.3. Разъем PS/2-Mouse

С интерфейсами Serial Mouse и PS/2-Mouse иногда возникают недоразумения. Хотя оба они последовательные, но имеют принципиальные различия в уровнях сигналов, способе синхронизации, частоте и формате посылок:

- Интерфейс PS/2 использует однополярный сигнал с уровнями ТТЛ, питание мыши — однополярное с напряжением +5 В относительно шины GND. Интерфейс RS-232C, применяемый в Serial Mouse, использует двухполярный сигнал (см. 2.1.1) с уровнями срабатывания +3 В и −3 В, требует двухполярного питания.
- Синхронный интерфейс PS/2-Mouse использует две раздельные сигнальные линии, одну — для передачи данных, другую — для сигналов синхронизации. Serial Mouse использует асинхронный способ передачи данных по одной линии.

Ясно, что совместимости между этими интерфейсами нет. Тем не менее, выпускаются и продаются переходники (пассивные!), позволяющие выбирать способ подключения мыши. Они предназначены только для *универсальных* мышей, у которых встроенный контроллер по напряжению питания способен распознать, к какому интерфейсу его подключили, и установить соответствующий тип своего выходного интерфейса. Универсальные мыши не особо распространены, поэтому часто приходится слышать о неудачных попытках применения таких переходников к обычным Serial Mouse или PS/2-Mouse.

Дополнительную путаницу вносят мыши для компьютеров Macintosh, которые имеют разъем, с виду напоминающий PS/2. Однако разъемы эти разные, да и интерфейс совершенно иной.

Манипулятор трекбол (*trackball*) («шар») представляет собой перевернутую мышь, шарик которой вращают пальцами. Иногда встраивается в клавиатуру.

3.3. Интерфейс игрового адаптера — GAME-порт

Интерфейс игрового адаптера занимает особое место в классификации. Он позволяет вводить значения *дискретных*

(4 бита) и *аналоговых* сигналов (сопротивления 4 резисторов). Изначально порт был предназначен для подключения джойстиков и других игровых устройств ввода (Paddle), но может применяться и для подключения «серьезных» датчиков. Метод измерения сопротивления основан на программном определении длительности импульса, пропорциональной сопротивлению. *Преобразование* начинается по выводу любого байта в регистр адаптера (201h), при этом биты 0–3 устанавливаются в 1. Время измеряется до возврата в нулевое состояние бит 0–3, соответствующих четырем аналоговым каналам. Если аналоговый вход закорочен на шину GND или цепь измеряемого сопротивления разорвана, соответствующий бит не обнулится. Поэтому в программе преобразования должен быть предусмотрен тайм-аут. Для измеряемых сопротивлений в диапазоне 0–100 кОм время определяется по формуле

$$T = 24{,}2 + 11 \times R,$$

где T — время (в микросекундах), а R — сопротивление (в килоомах).

Точность и линейность преобразования невысока, преобразование выполняется не быстро (до 1,12 мс) и сильно загружает процессор. Однако в отличие от «настоящих» аналого-цифровых преобразователей этот достается даром — игровой адаптер входит в состав практически всех комбинированных плат последовательных и параллельных портов, звуковых карт или системных плат.

Порт имеет разъем-*розетку DB-15S*. Назначение выводов и соответствие сигналов битам регистра приведены в табл. 3.3. Резисторы подключаются к шине питания +5 В, кнопки — к шине GND (рис. 3.4). Замыканию кнопок соответствуют нули в битах 5–7. Аналоговые каналы можно использовать для дискретного ввода, если их входы подключить к кнопкам, замыкающим их на шину GND, и к резисторам, «подтягивающим» их к уровню + 5 В. Два джойстика (А и В) подключаются через Y-образный переходник-разветвитель. На звуковых картах через разъем «*Game*» вместе с джойстиками могут подключаться и внешние MIDI-устройства через специальный кабель-адаптер, обеспечивающий гальваническую

развязку входного сигнала и ограничение выходного тока (см. рис. 2.16). Для интерфейса MIDI используются контакты 12 и 15, ранее предназначавшиеся для шин GND и +5V. Такое назначение делает безопасным подключение адаптера MIDI к «чистому» игровому порту и обычного джойстика к игровому порту с сигналами MIDI.

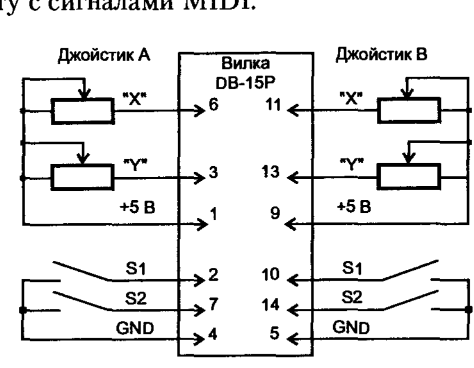


Рис. 3.4. Подключение датчиков к игровому адаптеру

Таблица 3.3. Интерфейс игрового адаптера и MIDI

Бит	Назначение	Контакт
7	Джойстик B кнопка #2	14
6	Джойстик B кнопка #1	10
5	Джойстик A кнопка #2	7
4	Джойстик A кнопка #1	2
3	Джойстик B Y-координата	13
2	Джойстик B X-координата	11
1	Джойстик A Y-координата	6
0	Джойстик A X-координата	3
–	GND	4, 5, (12)
–	+5 В	1, 8, 9, (15)
–	MIDI-Rx – вход (на звуковой карте)	15
–	MIDI-Tx – выход (на звуковой карте)	12

3.4. Аудиоинтерфейсы

В первых PC этот интерфейс был однонаправленным — традиционный канал управления звуком (PC Speaker) представлял собой примитивный синтезатор. Теперь для звуковоспроизведения (и звукозаписи) используется ставший стандартным двунаправленный цифровой аудиоканал. Роль PC Speaker сводится к подачам гудков при загрузке, идентификации ошибок во время POST, а также к сопровождению сообщений об ошибках.

3.4.1. PC Speaker

Стандартный канал управления звуком PC Speaker рассчитан на подключение высокоомного малогабаритного динамика. Логическая схема канала приведена на рис. 3.5. Звук формируется из тонального сигнала от второго канала системного таймера, которым можно программно управлять. Частоту сигнала можно изменять, программируя коэффициент деления счетчика-таймера, который расположен в пространстве ввода/вывода по адресу 042h. При записи (и чтении) 16-битный двоичный код передается парой (обязательно!) 8-битных операций (сначала младший байт, потом старший). Разрешая/запрещая формирование сигнала программно-управляемым битом 0 системного порта 61h, можно подавать сигналы определенной длительности. Такой способ формирования звука не загружает процессор и позволяет исполнять незамысловатые мелодии. С учетом инерции слуха быстрым переключением частот можно достигать эффекта многоголосия.

Более интересные звуки можно извлекать, используя принцип широтно-импульсной модуляции, программно осуществляемой через бит 1 порта 61h. В этом случае динамик выполняет роль фильтра низких частот (инерционного звена) демодулятора. Процессоры, начиная с 80286, способны формировать поток управляющих сигналов, который позволяет воспроизводить музыкальный или речевой сигнал с качеством карманного радиоприемника. Однако такое формирование звука практически полностью загружает процессор. Качество воспроизведения сильно зависит от частотных

свойств динамика. Предпочтительнее более крупные динамики, у которых лучше воспроизведение нижних частот, — с ними можно добиться даже разборчивости речи. Драйвер для звукоизвлечения существует также и для Windows 3.x/95, но в стандартную поставку Windows не входит.

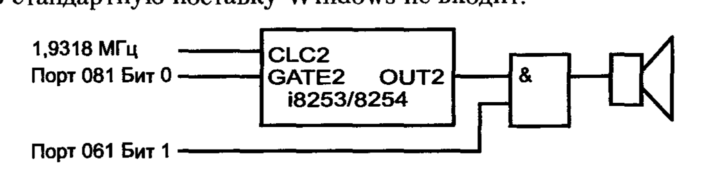


Рис. 3.5. Канал управления звуком

3.4.2. Цифровой аудиоканал

Кроме стандартного канала управления звуком, фактически стандартизованными являются средства работы с аудиосигналом, имеющиеся на плате Sound Blaster фирмы Creative Labs (табл. 3.4). Звуковые карты обычно имеют 16-битную шину ISA (8-битные нежелательны), PCI или PC Card, новые модели поддерживают PnP. Звуковая карта в своем составе имеет цифровой канал записи-воспроизведения моно- и стереофонического сигнала, микшер, синтезатор и MIDI-порт.

Цифровой аудиоканал обеспечивает возможность моно- или стереофонической записи и воспроизведения аудиофайлов с уровнем качества от кассетного магнитофона до аудио-CD. Запись производится оцифровкой сигнала с частотой дискретизации 5–44,1 кГц. Разрядность используемых ADC и DAC (аналого-цифровых и цифроаналоговых преобразователей) варьируется от 8 до 16 бит. Более совершенные схемы имеют встроенные средства компрессии сигнала, например, по методу адаптивной дельта-импульсно-кодовой модуляции (ADPCM). В этом случае в цифровом виде хранятся не значения выборок, а специальным образом обработанные разности величин соседних выборок, что позволяет (при сохранении качества) значительно уменьшить объем записи. Компрессия аудиосигнала применяется, например, в кодеках MPEG и голосовых модемах. На стандартных аудио-CD компрессия не применяется.

Для передачи потока данных по шине в цифровой аудиоканал и обратно используются 8- и/или 16-битные каналы DMA. Звуковой канал использует аппаратные прерывания и порты ввода/вывода. Для организации телефонной связи по сети (например, IPhone) нужен полнодуплексный режим (Full Duplex), реализуемый далеко не на всех картах.

Таблица 3.4. Назначение регистров Sound Blaster 16

Адрес*	R/W	Назначение
SB+0 (388h)	R	FM music status
SB+0 (388h)	W	FM music address register
SB+1 (389h)	W	FM music data register
SB+2 (38Ah)	W	Advanced FM music address register
SB+3 (38Bh)	W	Advanced FM music data register
SB+4	W	Mixer, I/O setup address register
SB+5	R/W	Mixer, I/O setup data register
SB+6	W	DSP reset
SB+8	W	FM music address register
SB+9	W	FM music data register
SB+0Ah	R	DSP read data
SB+0Ch	W	DSP write data or command
SB+0Ch	R	DSP write buffer status
SB+0Dh	R	DSP timer interrupt clear
SB+0Eh	R	DSP data available status
SB+0Fh	R	16-bit voice interrupt clear
SB+(10h...13h)		Reserved
MPU+0	R/W	MPU-401 data
MPU+1	R	MPU-401 status
MPU+1	W	MPU-401 command

* 38xh – адреса регистров FM music для плат Adlib.

SB – базовый адрес Sound Blaster (220h, 240h, 260h, 280h).

MPU – базовый адрес MIDI-порта (300h, 330h).

Оцифрованный звук хранится в файлах. Размер файла зависит от длительности записи, разрядности преобразования, частоты квантования и количества каналов (моно или стерео). Эти файлы редактируются ПО, которое позволяет вывести на экран подобие осциллограмм записанных сигналов.

Микшер с программным управлением обеспечивает регулировку входных и выходных сигналов, а также настройку тембра по низким и высоким частотам. Микшер позволяет смешивать входные сигналы от нескольких источников.

Синтезатор обеспечивает имитацию звучания музыкальных инструментов и воспроизведение различных звуков. В звуковых картах в основном используют два метода синтеза — частотный и волновой.

FM Music Synthesizer — синтезатор с частотной модуляцией (Frequency Modulated) имеет несколько каналов (голосов). Каждый канал содержит несколько синусоидальных генераторов (операторов), управляющих параметрами (частота, амплитуда) друг друга. Частотный синтезатор при небольшом числе операторов не обеспечивает хорошего синтеза.

Более совершенные платы имеют *WT Music Synthesizer* — синтезаторы на основе таблицы волн (Wave Table) — их называют также волновыми синтезаторами, — хранящие в своей постоянной памяти образцы некоторых звуков натуральных инструментов. При воспроизведении звука выходной сигнал пересчитывается встроенным процессором на соответствующие параметры частоты и амплитуды. Для хранения большего количества образцов возможна установка ОЗУ. Волновые синтезаторы обеспечивают высокое качество синтеза.

Для подключения внешних MIDI-устройств звуковые карты имеют порт *MIDI* (Musical Instrument Device Interface). Интерфейс — токовая петля с питанием от передатчика и гальванической развязкой входных цепей приемника (см. главу 2) — выведен на контакты 12, 15 разъема игрового адаптера. Обычно MIDI-порт эмулирует UART MPU-401, использующий два порта ввода/вывода и прерывание.

Для создания специальных эффектов (хор, реверберация и т. п.) на более сложных звуковых картах применяются процессоры обработки сигналов DSP (Data Signal Processor).

Звуковая карта имеет набор разъемов для подключения внешних сигналов (входные — микрофон, линейный вход, CD-ROM; выходные — линейный выход, выход на колонки или наушники), некоторые платы имеют цифровой вход от CD-ROM. Встроенный усилитель имеет мощность до 4 Вт на канал. Колонки *(Speakers)* для PC имеют специальный магнитный экран или улучшенную конструкцию магнитной системы динамиков, чтобы предотвратить воздействие магнитного поля на монитор. Активные колонки *(Active Speakers)* имеют встроенный усилитель и требуют внешнего питания. Полоса частот колонок PC недостаточна для Hi-Fi воспроизведения. Более качественные системы имеют две колонки для средних и высоких частот и одну большую (сабвуфер) — для низких.

Для выдачи MIDI-команд существуют специальные *MIDI-клавиатуры*, простейшие фиксируют и передают только факт нажатия-отпускания клавиш, более сложные имеют динамические датчики, реагирующие на силу и скорость нажатия. Динамическая клавиатура в сочетании с хорошим WT-синтезатором обеспечивает достаточно полную имитацию фортепиано и других инструментов. MIDI-интерфейс имеют многие профессиональные и полупрофессиональные клавишные синтезаторы.

Подключение к звуковой карте внешних устройств осуществляется через малогабаритные стереоразъемы — «джеки» (jack) — на задней панели:

- *Line In* — линейный вход от магнитофона, тюнера, проигрывателя, синтезатора и т. п. Чувствительность порядка 0,1–0,3 В.
- *Mic In* — микрофонный вход, чувствительность 3–10 мВ.
- *Line Out* — линейный выход сигнала на внешний усилитель или магнитофон, уровень сигнала порядка 0,1–0,3 В.
- *Speaker Out* — выход на акустические системы или головные телефоны. Подключать к нему внешний усилитель

мощности нецелесообразно, поскольку здесь искажения больше, чем на линейном выходе.

- *Joystick/MIDI* – разъем подключения джойстика. Для подключения MIDI-устройств необходим кабель-адаптер (см. рис. 2.16).

Для подключения внутренних устройств CD-ROM к аналоговым входам используются четырехштырьковые разъемы, различающиеся как шагом между выводами, так и их назначением. Часто ставят рядом несколько разъемов с параллельно соединенными сигнальными контактами. Но это может не сработать, если кабель имеет другое расположение сигналов. Тогда можно переустановить контакты на разъеме кабеля, для чего иголкой нажимают на фиксирующий выступ контакта, после чего контакт вытягивают в сторону кабеля и переставляют в другое гнездо. Сигнальные контакты аудиовходов см. на рис. 3.6. Разъем может иметь ключ с противоположной стороны (по ошибке сборщика кабеля или по внутреннему стандарту его производителя). Задача подключения облегчается тем, что требует правильной расстановки только двух сигнальных контактов. Контакты общего провода выделяются тем, что на плате соединяются с шиной, а на кабеле – с экраном. Положение левого и правого каналов аудио-CD в большинстве случаев непринципиально.

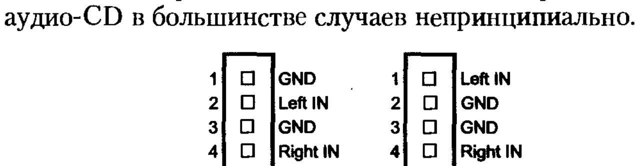


Рис. 3.6. Внутренние разъемы подключения аудиосигналов

3.5. Интерфейсы видеомониторов

В традиционной технике цветного телевизионного вещания (PAL, SECAM или NTSC) видеосигнал непосредственно несет информацию о мгновенном значении яркости, а цветовая информация передается в модулированном виде на дополнительных частотах. Таким образом обеспечивается совместимость черно-белого приемника, игнорирующего цве-

товую информацию, с цветным передающим каналом. Однако для вывода графической информации с высоким разрешением ни одна из традиционных вещательных систем не подходит, поскольку они имеют существенно ограниченную полосу пропускания цветовых каналов.

Для графики низкого разрешения, при которой частоты синхронизации были близки к стандартным телевизионным, использовался интерфейс *Composite Video.* По коаксиальному кабелю (75 Ом) передавался полный стандартный видеосигнал с размахом около 1,5 В. В видеотехнике этот сигнал называют низкочастотным, имея в виду то, что по нему подается прямой, а не модулированный сигнал. Соответствующий ему вход имеется не у каждого телевизора. При наличии дополнительного радиочастотного модулятора *RFM* (Radio Frequency Modulator) можно было подключаться к антенному входу стандартного телевизионного приемника, но при этом еще более снижалось реальное разрешение графики. Для композитного интерфейса используют коаксиальные разъемы RCA («колокольчик»), широко применяемые в видео- и аудиотехнике.

Для мониторов при высоком разрешении можно использовать только прямую подачу сигнала на входы видеоусилителей базовых цветов — *RGB-вход* (*Red Green Blue* — красный, зеленый и синий). Интерфейс между видеоадаптером и монитором может быть как дискретным (с сигналами ТТЛ), так и аналоговым.

3.5.1. Дискретный интерфейс RGB TTL

Первые мониторы для PC имели цифровой интерфейс с уровнями ТТЛ (табл. 3.5) — *RGB TTL*. Для монохромного монитора использовали лишь два сигнала — видео (включить/выключить луч) и повышенной яркости. Таким образом, монитор мог отобразить три градации яркости: хотя 2^2=4, но «темный пиксел» и «темный с повышенной яркостью» неразличимы. В цветных мониторах класса *CD* (*Color Display*) имелось по одному сигналу для включения каждого луча и общий сигнал повышенной яркости. Таким образом, можно было задать 16 цветов. Следующий класс — улучшенный цветной дисплей *ECD* (*Enhanced Color Display*)

имел цифровой интерфейс с двумя сигналами на каждый базовый цвет. Сигналы позволяли задавать одну из 4 градаций интенсивности; общее количество кодируемых цветов достигло 2^6=64. Сигналы RED, GREEN, BLUE и Red, Green, Blue обозначают, соответственно, старшие и младшие биты базисных цветов.

Строчная и кадровая синхронизация монитора осуществляется сигналами H.Sync и V.Sync. Монохромные адаптеры MDA и HGC, работающие с высоким разрешением (720×350 пикселов), используют высокую частоту развертки. Адаптер CGA работает с низкими частотами (параметры синхронизации близки к телевизионным). Адаптеры и мониторы EGA могут работать с любыми из этих частот. Для облегчения переключения режимов генератора развертки монитора используют сигнал V.Sync: полярность импульсов определяет диапазон частот развертки текущего видеорежима. Для всех разновидностей интерфейса RGB TTL используется разъем DB-9S.

Таблица 3.5. Цифровой интерфейс монитора (RGB TTL)

Контакт	Монитор		
	MDA/HGC	**CGA**	**EGA Color/Mono**
1	GND	GND	GND
2	GND	GND	Red
3	–	RED	RED
4	–	GREEN	GREEN
5	–	BLUE	BLUE
6	Intensiv.	Intensiv.	Green/Intens.
7	Video	Reserved	Blue/Video
8	+H.Sync.	+H.Sync.	+H.Sync.
9	–V.Sync.	+V.Sync.	–(+)V.Sync.

3.5.2. Аналоговый интерфейс RGB

Из-за ограничений цифрового интерфейса пришлось перейти на аналоговый интерфейс, перенеся цифроаналоговые

преобразователи сигналов базовых цветов из монитора на графический адаптер. Такой интерфейс с 8-разрядными ЦАП для каждого цвета в настоящее время позволяет выводить 16,7 миллиона цветов (*True Color*). Этот интерфейс называется *RGB Analog*, в нем базовые цвета передаются аналоговыми сигналами с отдельными обратными линиями по витым парам. Черному цвету соответствует нулевой потенциал на линиях всех цветов, полной яркости каждого цвета соответствует уровень +0,7 В. Сигналы управления, состояния и синхронизации передаются сигналами ТТЛ. Впервые аналоговый интерфейс был применен на адаптере PGA фирмы IBM, где для него использовался 9-контактный разъем DB-9S (табл. 3.6). В дальнейшем, начиная с адаптеров VGA, стали применять малогабаритный 15-контактный разъем с таким же внешним размером (табл. 3.7). По назначению сигналов эти интерфейсы в основном совпадают, и существуют даже переходные кабели с 15- на 9-контактные разъемы (табл. 3.8). В адаптере PGA используется *совмещенная синхронизация* (*Composite Sync*) сигналом (H+V)Sync., этот режим поддерживают и многие современные мониторы.

Таблица 3.6. Аналоговый интерфейс монитора PGA (разъем DB-9S)

Контакт	Сигнал
1	Red
2	Green
3	Blue
4	(H+V)Sync
5	Mode Control
6	Red Return
7	Green Return
8	Blue Return
9	GND

Таблица 3.7. Аналоговый интерфейс монитора (RGB Analog)

Контакт DB-15	Видеоадаптер MCGA/VGA/SVGA/XGA	Монитор Mono	Монитор Color
1	Red	–	Red
2	Green	Video	Green
3	Blue	–	Blue
4	ID2	–	–
5	GND/DDC Return[1]	SelfTest/DDC Return	SelfTest/DDC Return
6	Red Return	Key	Red Return
7	Green Return	Video Return	Green Return
8	Blue Return	–	Blue Return
9	Ключ (нет контакта)	–	–
10	GND	GND	GND
11	ID0	–	GND
12	ID1/SDA[1]	– /SDA	GND/SDA
13	H.Sync/(H+V)Sync[2]	H.Sync/(H+V)Sync	H.Sync/(H+V)Sync
14	V.Sync	V.Sync	V.Sync
15	SCL[1]	SCL	SCL

[1] Сигналы DDC Return, SDA и SCL используются только при поддержке DDC. При этом контакт 9 может использоваться для питания логики монитора.

[2] Сигнал (H+V)Sync используется при совмещенной синхронизации (Composite Sync).

Таблица 3.8. Переходник 9–15 аналогового интерфейса монитора

Контакт DB9	Сигнал	Контакт DB15
1	Red	1
2	Green	2
3	Blue	3
4	H.Sync	13
5	V.Sync	14
6	Red Return	6
7	Green Return	7
8	Blue Return	8
9	GND	10, 11

Несмотря (в прямом смысле) на наличие ключа — D-образного кожуха — 15-контактные разъемы ухитряются вставлять в перевернутом положении, при этом один из контактов среднего ряда подгибается, а потом и ломается (штырьки этих разъемов тоньше и слабее, чем у 9-контактных). Естественно, монитор, подключенный таким образом, работать не будет.

В компьютерах Macintosh монитор, совместимый по параметрам с VGA, имеет разъем DB-15P (такой же, как и у Game-порта PC). Назначение его выводов приведено в табл. 3.9.

Таблица 3.9. Разъем VGA Macintosh

Контакт	Сигнал
1	Red Return
2	Red
3	Comp.Sync
4	ID0
5	Green
6	Green Return
7	ID1
8	Не используется
9	Blue
10	ID2
11	Sync. GND
12	V.Sync
13	Blue Return
14	H.Sync GND
15	H.Sync

Кроме передачи изображения по интерфейсу передают информацию, необходимую для автоматизации согласования параметров и режимов монитора и компьютера. «Интересы» компьютера представляет дисплейный адаптер, к которому и подключается монитор. С его помощью обеспечиваются идентификация монитора, необходимая для PnP, и управление энергопотреблением монитора.

Для простейшей *идентификации* в интерфейс ввели три логических сигнала ID0–ID2, по которым адаптер мог определить тип подключенного монитора IBM (табл. 3.10). Со стороны монитора эти линии либо подключались к шине GND, либо оставлялись неподключенными. Однако из этой системы идентификации использовали лишь сигнал ID1, по которому определяют подключение монохромного монитора. Монохромный монитор может быть опознан адаптером и иначе — по отсутствию нагрузки на линиях Red и Blue.

Таблица 3.10. Параллельная идентификация мониторов IBM

Дисплей	ID0	ID1	ID2
Монохромный 12" IBM8503	NC	GND	NC
Цветной 12" IBM8513	GND	NC	NC
Цветной 14" IBM8512	GND	NC	NC
Цветной 15" IBM8514	GND	NC	GND

Параллельную идентификацию мониторов заменила последовательная по каналу цифрового интерфейса *VESA DDC* (Display Data Channel). Этот канал построен на интерфейсе I^2C (DDC2B) или *ACCESS Bus* (DDC2AB), которые используют всего два ТТЛ-сигнала SCL и SDA. Интерфейс *DDC1* является однонаправленным — монитор посылает адаптеру блок своих параметров по линии SDA (контакт 12), которые синхронизируются сигналом V.Sync (контакт 14). На время приема блока параметров адаптер может повысить частоту V.Sync до 25 кГц (генератор кадровой развертки по такой высокой частоте синхронизироваться не будет). Интерфейс *DDC2* является двунаправленным; для синхронизации используется выделенный сигнал SCL (контакт 15). Интерфейс *DDC2AB* отличается тем, что допускает подключение ПУ, не требующих высокой скорости обмена, к компьютеру по последовательной шине *ACCESS Bus*. При этом внешний разъем шины выносится на монитор (см. табл. 7.3).

Блок параметров расширенной идентификации дисплея *EDID* (Extended Display Identification) имеет одну и ту же структуру для любой реализации DDC (табл. 3.11).

Таблица 3.11. Блок расширенной идентификации EDID

Смещение, байт	Длина, байт	Назначение
0	8	Заголовок (индикатор начала потока EDID)
8	10	Идентификатор изделия (назначается производителем)
18	2	Версия EDID
20	15	Основные параметры и возможности дисплея
35	19	Установленные параметры синхронизации
54	72	Дескрипторы параметров синхронизации (байты 4–18)
126	1	Флаг расширения
127	1	Контрольная сумма

Для *управления энергопотреблением* монитора в соответствии со стандартом VESA *DPMS* (Display Power Management Signaling) используются сигналы кадровой и строчной синхронизации V.Sync и H.Sync (табл. 3.12).

Таблица 3.12. Управление энергопотреблением монитора (VESA DPMS)

Режим	H.Sync	V.Sync
On	Активен	Активен
Standby	Неактивен	Активен
Suspend	Активен	Неактивен
Off	Неактивен	Неактивен

Разъемы, применяемые в современных адаптерах и мониторах SVGA, не предназначены для передачи высокочастотных сигналов. Пределом для них является примерно 150 МГц, что для высокого разрешения и высокой частоты регенерации недостаточно. Поэтому на больших профессиональных мониторах с высокими разрешением и частотами синхронизации и на соответствующих адаптерах имеются *BNC-разъемы* для соединения с помощью коаксиальных кабелей.

3.5.3. Комбинированный интерфейс EVC

Для расширения частотного диапазона и, учитывая тенденцию к использованию последовательных шин USB и FireWire, для подключения ПУ к системному блоку компьютера VESA предложила новый тип разъема *EVC* (Enhanced Video Connector). Кроме обычного аналогового интерфейса RGB и канала DDC2 разъем EVC имеет контакты для видеовхода, входные и выходные стереоаудиосигналы, шины USB и FireWire, а также линии питания постоянного тока для зарядки аккумуляторов портативных ПК. Разъем имеет две секции: высокочастотную для присоединения четырех коаксиальных кабелей и низкочастотную на 30 контактов (рис. 3.7). Контакты высокочастотной секции, хотя и не являются коаксиальными, позволяют передавать сигналы с частотами до 2 ГГц. Контактом экранов является крестообразная перегородка. При использовании 75-омных коаксиальных кабелей на частоте 500 МГц гарантируется уровень отражений и перекрестных помех не выше 2%. Контакты C1, C2 и C4 используются для передачи цветовых сигналов R, G и B соответственно, контакт C3 служит для передачи синхросигнала пикселов. Низкочастотная секция описана в табл. 3.13. Разъем поделен на компактные зоны для каждой группы сигналов, правда, шины USB и 1394 используют общий контакт для экрана. Назначение контактов видеовхода (S-Video или композитный, PAL или NTSC) может программироваться по каналу DDC2.

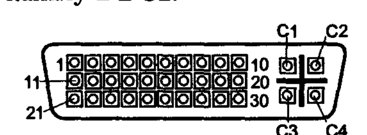


Рис. 3.7. Разъем EVC (розетка)

Стандарт определяет три уровня реализации: базовый, мультимедийный и полный. Базовый включает только видеосигналы и DDC, в мультимедийном должны быть аудиосигналы. При использовании коннектора в полном объеме монитор превращается в коммутационный центр, который соединяется с компьютером одним кабелем, а все остальные ПУ

(включая клавиатуру, мышь, принтер) подключаются к монитору. Разъем может использоваться для подключения портативного ПК к док-станции. EVC собирает сигналы от разных подсистем — графической, видео, аудио, последовательных шин и питания. Этот общий разъем, устанавливаемый на корпусе системного блока, может соединяться с разными платами внутренними кабелями через промежуточные разъемы.

Таблица 3.13. Назначение контактов низкочастотной части EVC

Контакт	Цепь	Контакт	Цепь	Контакт	Цепь
1	Audio Output, Right	11	Charging power input, +	21	Audio input, left
2	Audio Output, Left	12	Charging power input, –	22	Audio input, right
3	Audio Output, Return	13	Video input, Y or composite in	23	Audio input, return
4	Sync Return	14	Video input, return	24	Stereo sync (TTL)
5	Horizontal Sync (TTL)	15	Video input, C in	25	DDC return
6	Vertical Sync (TTL)	16	USB data +	26	DDC data (SDA)
7	RESERVED 1	17	USB data –	27	DDC, clock (SCL)
8	RESERVED 2	18	USB/1394 common mode shield	28	+5 В
9	1394 pair A, data –	19	1394 Vg	29	1394 pair B, clock +
10	1394 pair B, data +	20	1394 Vp	30	1394 pair B, clock –

4. Шины расширения

Шины расширения (Expansion Bus) предназначены для подключения различных адаптеров ПУ. Интерфейсы шин расширения PC ведут историю с 8-битной шины *ISA*. Ее открытость обеспечила появление широкого спектра плат расширений, позволивших использовать PC в различных сферах. С появлением AT-286 шина была расширена по разрядности и возможностям. Шина *EISA* была призвана сделать обмен еще более производительным и надежным. Она содержала прогрессивные идеи автоматизации конфигурирования (прототип PnP), позволяя устанавливать и ISA-адаптеры. Шина *MCA*, предложенная IBM, не была поддержана производителями PC, так как ее спецификация была закрытой. С появлением процессора 486 родилась высокоскоростная локальная шина *VLB*. Однако она являлась дополнением к слоту шины ISA/EISA и использовалась в основном лишь для графических карт и дисковых контроллеров. Принципиальная привязка к шине процессора 486 не обеспечила ей долголетия. Современная скоростная шина *PCI* является стандартной для компьютеров с процессорами x86 всех поколений старше четвертого, она используется в Power PC и других платформах. Развитием шины PCI, нацеленным на дальнейшее ускорение обмена, явился порт *AGP*, предназначенный для подключения графических адаптеров. Для блокнотных компьютеров, поначалу имевших закрытую архитектуру, потребность в подключении периферии привела к появлению стандартизованной шины *PCMCIA*, впоследствии переименованной в *PC Card*.

Шины расширения системного уровня позволяют адаптерам максимально использовать системные ресурсы PC: пространства памяти и ввода/вывода, прерывания, каналы прямого доступа к памяти. Как следствие, изготовителям модулей расширения приходится точно следовать протоколам шины, выдерживая жесткие частотные и нагрузочные параметры, а

также временные диаграммы. Отклонения приводят к несовместимости с некоторыми системными платами. Если при подключении к внешним интерфейсам это приведет к неработоспособности только самого устройства, то некорректное подключение к системной шине может блокировать работу всего компьютера. Следует также учитывать ограниченность ресурсов PC. Самые дефицитные из них — линии запросов прерываний (каналы прямого доступа можно заменить на активное управление шиной PCI). Проблемы распределения ресурсов на шинах решаются по-разному, но чаще всего применяется технология PnP.

В табл. 4.1 даны характеристики стандартных шин расширения PC.

Таблица 4.1. Характеристики шин расширения

Шина	Пропускная способность, Мбайт/с*	Каналы DMA	Bus-Master	ACFG**	Разрядность данных	Разрядность адреса	Частота, МГц
ISA-8	4	3	–	–	8	20 (1 Мбайт)	8
ISA-16	8/(16)	8	+	–	16	24 (16 Мбайт)	8/(16)
EISA	33,3	8	+	+	32	32 (4 Гбайт)	8,33
MCA-16	16	–	+	+	16	24 (16 Мбайт)	10
MCA-32	20	–	+	+	32	32 (4 Гбайт)	10
VLB	132	–	(+)	–	32/64	32 (4 Гбайт)	33–50(66)
PCI	132/264	–	–	+	32/64	32 (4 Гбайт)	33(66)
PCMCIA	–	–	–	+	16	26 (64 Мбайт)	33

* Указана максимальная пропускная способность. Реальная примерно в 2 раза ниже за счет прерываний, регенерации и протокольных процедур.

** Поддержка автоматнческого конфигурирования. Для ISA PnP является позднейшей надстройкой, реализуемой адаптерами и ПО.

Шины расширения конструктивно оформляются в виде щелевых разъемов (слотов) на системной плате для установки плат адаптеров. Унификация системных плат, корпусов и плат расширения обеспечивается:

- стандартизацией размеров, количества контактов и электрического интерфейса слотов шин расширения;
- фиксированным расстоянием от слота до задней кромки платы;

- фиксированным шагом между соседними слотами, а также их привязкой к крепежным точкам и разъему клавиатуры;
- определением максимальных габаритов (длины и высоты) карт расширения;
- определением геометрии нижнего края платы расширения, формы и размера фиксирующей скобки.

Размеры карт расширения для самой популярной шины — ISA — приведены на рис. 4.1.

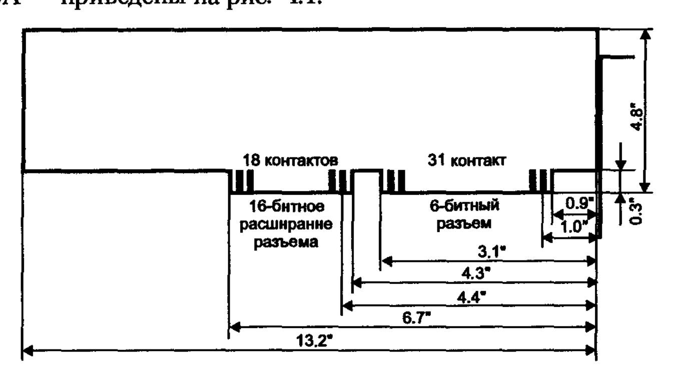


Рис. 4.1. Карта расширения для шины ISA

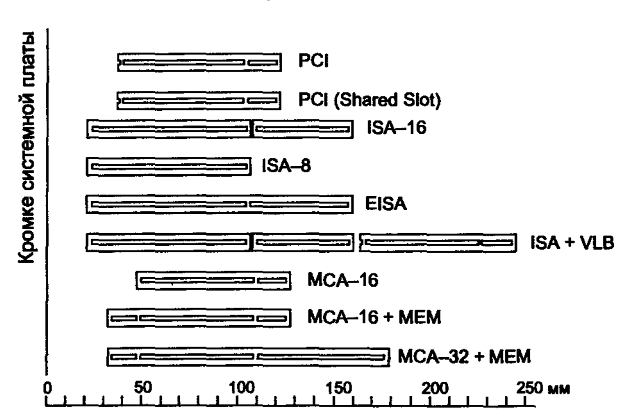


Рис. 4.2. Вид и положение слотов шин расширения

Типы слотов легко определить визуально (рис. 4.2). На этом рисунке присутствие всех типов шин условно — реально на

системных платах находится не более двух-трех типов слотов. Распространенные сочетания: ISA+PCI, ISA+VLB, EISA+PCI, EISA+VLB. Фирмой ASUSTek применяется слот «Media BUS», дополняющий слот PCI сигналами шины ISA.

У адаптеров для шины PCI, в отличие от ISA/EISA и VLB, компоненты расположены на левой стороне печатной платы. Для экономии места на системной плате используют так называемый *разделяемый слот* (Shared Slot). На самом деле это разделяемое окно на задней стенке корпуса, которое может использоваться либо картой ISA, либо картой PCI. Таким образом, максимальное суммарное количество доступных слотов оказывается на единицу меньшим, чем видимое количество.

Для *низкопрофильных корпусов* системные платы имеют всего один слот расширения, в который устанавливается специальная плата-переходник *Riser Card*. Этот переходник специфичен для модели системной платы (корпуса), поскольку на его краевой разъем заводятся линии нескольких системных шин. Если Riser Card имеет слоты только шины ISA, он обычно вставляется в стандартный слот ISA-16. С точки зрения уменьшения наводок и паразитных емкостей лучше использовать специальные платы с одним разъемом, чтобы не перегружать шину неиспользуемыми проводниками и разъемами.

Шины расширения допускают конфигурирование, которое предполагает в основном настройку их временных параметров:

- Для шины VLB применяется перемычка, управляющая делителем частоты сигнала синхронизации в зависимости от того, превышает ли системная частота 33,3 МГц.
- Для шины PCI частота синхронизации определяется частотой системной шины процессора. Кроме того, режимы могут определяться в BIOS Setup.
- Для шины ISA кроме частоты (которая должна быть порядка 8 МГц) задают время восстановления для 8- и 16-битных обращений к памяти и вводу/выводу. Неустойчивая работа адаптеров может потребовать замедления шины ISA. Понижение ее производительности не отражается на общей производительности современных компьютеров.

- Иногда для шин ISA и PCI опциями BIOS Setup приходится распределять системные ресурсы (линии запросов прерываний)(см. приложение А).

4.1. Шины ISA, EISA и PC-104

ISA Bus (Industry Standard Architecture) — шина расширения, применявшаяся с первых моделей PC и ставшая промышленным стандартом. В компьютере XT использовалась шина с разрядностью данных 8 бит и адреса — 20 бит. В компьютерах AT ее расширили до 16 бит данных и 24 бит адреса. В таком виде она существует и поныне. Конструктивно шина выполнена в виде двух щелевых разъемов с шагом выводов 2,54 мм (0,1 дюйма) (см. рис. 4.3). Подмножество ISA-8 использует только 62-контактный слот (ряды A, B), в ISA-16 применяется дополнительный 36-контактный слот (ряды C, D).

Шина обеспечивает возможность отображения 8- или 16-битных регистров на пространство ввода/вывода и памяти. Диапазон адресов памяти ограничен областью UMA (выше A0000h). Для шины ISA-16 опциями BIOS Setup может быть разрешено пространство между 15-м и 16-м мегабайтами памяти (при этом компьютер не сможет использовать более 15 Мбайт ОЗУ). *Диапазон адресов ввода/вывода* сверху ограничен количеством используемых для дешифрации бит адреса, нижняя граница ограничена областью адресов 0—FFh, зарезервированных под устройства системной платы. В PC была принята 10-битная адресация ввода/вывода, при которой линии адреса A[15:10] устройствами игнорировались. Таким образом, диапазон адресов устройств шины ISA ограничивается областью 100h—3FFh, то есть всего 758 адресов 8-битных регистров. На некоторые из адресов претендуют системные устройства (см. табл. A.1). Впоследствии стали применять 12-битную адресацию (диапазон 100h—FFFh). При ее использовании необходимо учитывать возможность присутствия на шине старых 10-битных адаптеров, которые «отзовутся» на адрес с подходящими ему битами A[9:0] во всей допустимой области 12-битного адреса четыре раза (у каждого 10-битного адреса будет еще по три 12-битных псевдонима).

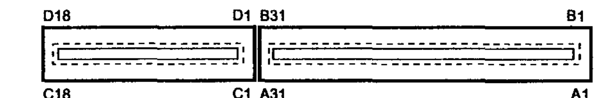


Рис. 4.3. Слот ISA

Шина ISA-8 может предоставить до 6 линий запросов прерываний, ISA-16 — 11. Часть их могут «отобрать» устройства системной платы или шина PCI.

Шина позволяет использовать до трех 8-битных *каналов DMA*. На 16-битной шине доступны еще три 16-битных канала. Сигналы 16-битных каналов могут использоваться для прямого управления шиной устройством *Bus-Master*. При этом канал DMA применяется для арбитража шины, а адаптер Bus-Master формирует все адресные и управляющие сигналы шины, не забывая передать управление шиной процессору не позднее, чем через 15 мкс (чтобы не нарушить регенерацию памяти).

Все перечисленные ресурсы системной шины должны быть бесконфликтно распределены. Бесконфликтность подразумевает следующее:

- Каждое устройство должно управлять шиной данных только при чтении по своим адресам или по используемому им каналу DMA. Области адресов для чтения не должны пересекаться. «Подсматривать» операции записи, «адресованные не ему», не возбраняется.
- Назначенную линию IRQx или DRQx устройство должно держать на низком уровне в пассивном состоянии и переводить в высокий уровень для активации запроса. Неиспользуемыми линиями запросов устройство управлять не имеет права, они должны быть электрически откоммутированы или подключаться к буферу, находящемуся в третьем состоянии. Одной линией запроса может пользоваться только одно устройство. Такая нелепость (с точки зрения схемотехники ТТЛ) была допущена в первых PC и из требований совместимости тиражируется до сих пор.

Задача распределения ресурсов для старых адаптеров решалась с помощью джамперов, затем появились программно конфигурируемые устройства, которые вытесняются автоматически конфигурируемыми платами PnP.

С появлением 32-битных процессоров делались попытки расширения разрядности шины, но все 32-битные шины ISA не являются стандартизованными, кроме EISA.

EISA Bus (Extended ISA) — стандартизованное расширение ISA до 32 бит. Конструктивное исполнение обеспечивает совместимость с ней ISA-адаптеров (рис. 4.4). Узкие дополнительные контакты расширения (ряды E, F, G, H) расположены между ламелями (контактными площадками) разъема ISA и ниже ламелей A, B, C, D таким образом, что адаптер ISA, не имеющий дополнительных ключевых прорезей в краевом разъеме, не достает до них. Установка карт EISA в слоты ISA недопустима, поскольку ее цепи попадут на контакты ISA, в результате чего системная плата окажется неработоспособной (к счастью, «без дыма»).

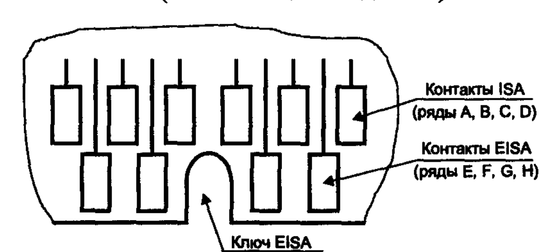


Рис. 4.4. Разъем шины EISA

Для режимов EISA используются дополнительные управляющие сигналы, обеспечивающие более эффективные режимы передачи. В обычном (не пакетном) режиме передачи за каждую пару тактов может быть передано до 32 бит (один такт на фазу адреса, один — на фазу данных). В *пакетном режиме* (Burst Mode) пакеты данных пересылаются без указания текущего адреса внутри пакета. В пакете очередные данные передаются в каждом такте шины, длина пакета достигает 1024 байт. Шина предусматривает более производительные режимы DMA (см. прил. A.4), при которых скорость обмена может достигать 33 Мбайт/с. Линии запросов прерываний допускают разделяемое использование, причем сохраняется совместимость с ISA-картами: каждая линия запроса может программироваться на чувствительность как по высокому (как в ISA), так и по низкому уровню. Шина до-

пускает потребление каждой картой расширения мощности до 45 Вт, но это не означает, что мощность блока питания для 8 слотов должна быть более 360 Вт — полную мощность не потребляет ни один из адаптеров.

Многие решения EISA напоминают MCA. Каждый слот (максимум — 8) и системная плата имеют селективное разрешение адресации ввода/вывода и отдельные линии запроса и подтверждения управления шиной. Арбитраж запросов выполняет устройство *ISP* (Integrated System Peripheral). Приоритеты (в порядке убывания): регенерация, DMA, CPU, Bus-Master. Обязательной принадлежностью системной платы с шиной EISA является *энергонезависимая память конфигурации NVRAM*, в которой хранится информация об устройствах EISA для каждого слота. Формат записей стандартизован, для модификации информации применяется специальная утилита *ECU* (EISA Configuration Utility). Архитектура позволяет при использовании программно конфигурируемых адаптеров автоматически разрешать конфликты, но в отличие от PnP не допускает динамического конфигурирования — после выхода из режима конфигурирования необходима перезагрузка. Изолированный доступ к портам ввода/вывода каждой карты во время конфигурирования обеспечивается просто: сигнал AEN, разрешающий декодирование адреса в цикле ввода/вывода, на каждый слот приходит по отдельной линии AENx, в это время программно-управляемой. Так можно обращаться и к картам ISA, но из этого нельзя извлечь выгоды, поскольку они не поддерживают обмена конфигурационной информацией. Идеи конфигурирования EISA были использованы в спецификации PnP для шины ISA (формат конфигурационных записей ESCD напоминает NVRAM EISA).

EISA — дорогая, но оправдывающая себя архитектура, применяющаяся в многозадачных системах и на серверах, где требуется высокоэффективное расширение шины ввода/вывода. Перед шиной PCI у нее есть преимущество в количестве слотов (8 против 4).

Назначение контактов слотов шин ISA и EISA приведено в табл. 4.2 и 4.3.

Таблица 4.2. Разъемы шин ISA-8, ISA-16 и EISA

Ряд F	Ряд B	N#	Ряд A	Ряд E
GND	GND	1	IOCHK	CMD#
+5 В	Reset	2	Data 7	START#
+5 В	+5 В	3	Data 6	EXRDY
Unused	IRQ 2/9[1]	4	Data 5	EX32#
Unused	–5 В	5	Data 4	GND
Key	DRQ 2	6	Data 3	Key
Unused	–12 В	7	Data 2	EX16#
Unused	0WS#[2]	8	Data 1	SLBURST#
+12 В	+12 В	9	Data 0	MSBURST#
M/–IO	GND	10	IOCHDRY	W/R#
–LOCK	SmemWR#	11	AENx	GND
Reserved	SmemRD#	12	Addr 19	EMB66#[3]
GND	IO WR#	13	Addr 18	EMB133#[3]
Reserved	IO RD#	14	Addr 17	Reserved
BE 3#	DACK 3#	15	Addr 16	GND
Key	DRQ 3	16	Addr 15	Key
BE 2#	DACK 1#	17	Addr 14	BE 1#
BE 0#	DRQ 1	18	Addr 13	LA 31#
GND	Refr#	19	Addr 12	GND
+5 В	Bclock	20	Addr 11	LA 30#
LA 29#	IRQ 7	21	Addr 10	LA 28#
GND	IRQ 6	22	Addr 9	LA 27#
LA 26#	IRQ 5	23	Addr 8	LA 25#
LA 24#	IRQ 4	24	Addr 7	GND
Key	IRQ 3	25	Addr 6	Key
LA 16	DACK 2#	26	Addr 5	LA 15
LA 14	TC	27	Addr 4	LA 13
+5 В	BALE	28	Addr 3	LA 12
+5 В	+5 В	29	Addr 2	LA 11
GND	Osc.	30	Addr 1	GND
LA 10	GND	31	Addr 0	LA 9

[1] B4: XT=IRQ2, AT=IRQ9.

[2] B8: XT=Card Selected.

[3] E12, E13 — только для Fast EISA.

Таблица 4.3. Дополнительные разъемы шин ISA-16 и EISA

Ряд H	Ряд D	N#	Ряд C	Ряд G
LA 8	MCS16#	1	SBHE#	LA 7
LA 6	IOCS16#	2	LA 23	GND
LA 5	IRQ 10	3	LA 22	LA 4
+5 В	IRQ 11	4	LA 21	LA 3
LA 2	IRQ 12	5	LA 20	GND
Key	IRQ 15	6	LA 19	Key
Data 16	IRQ 14	7	LA 18	Data 17
Data 18	DACK0#	8	LA 17	Data 19
GND	DRQ 0	9	MemRD#	Data 20
Data 21	DACK5#	10	MemWR#	Data 22
Data 23	DRQ 5	11	Data 8	GND
Data 24	DACK6#	12	Data 9	Data 25
GND	DRQ 6	13	Data 10	Data 26
Data 27	DACK7#	14	Data 11	Data 28
Key	DRQ 7	15	Data 12	Key
Data 29	+5 В	16	Data 13	GND
+5 В	Master#	17	Data 14	Data 30
+5 В	GND	18	Data 15	Data 31
MAKx#	–	19	–	MREQx

Сигналы шины ISA восходят к шинам Microbus и Multibus, они естественны для периферийных микросхем фирмы Intel для процессоров 8080 и 80x86/88. *Набор сигналов* ISA-8 предельно прост. *Программное обращение* к ячейкам памяти и пространства ввода/вывода обеспечивают следующие сигналы:

- Data[7:0] — шина данных.
- Addr[19:0] — шина адреса.
- AEN — разрешение адресации портов (запрещает ложную дешифрацию адреса в цикле DMA).
- IOWR# — запись в порт.
- IORD# — чтение порта.

- SMemWR# — запись в память (в диапазоне адресов 0–FFFFFh).
- SMemRD# — чтение памяти (в диапазоне адресов 0–FFFFFh).

К сигналам *запросов прерывания* и *каналам прямого доступа к памяти* относятся следующие:

- IRQ2/9, IRQ[3:7] — запросы прерываний. Положительный перепад сигнала вызывает запрос аппаратного прерывания. Для идентификации источника высокий уровень должен сохраняться до подтверждения прерывания процессором, что затрудняет разделяемое использование прерываний. Линия IRQ2/9 в шинах XT вызывает аппаратное прерывание с номером 2, а в AT — с номером 9.
- DRQ[1:3] — запросы 8-битных каналов DMA (положительным перепадом).
- DACK[1:3]# — подтверждение запросов 8-битных каналов DMA.
- TC — признак завершения счетчика циклов DMA.

Шина имеет и несколько служебных сигналов *синхронизации, сброса и регенерации памяти*, установленной на адаптерах:

- IOCHRDY — готовность устройства, низкий уровень удлиняет текущий цикл (не более 15 мкс).
- BALE — разрешение защелки адреса. После его спада в каждом цикле процессора линии Addr 0–19 гарантированно содержат действительный адрес.
- Refr# — цикл регенерации памяти (в XT называется DACK 0#). Сигнал появляется каждые 15 мкс, при этом шина адреса указывает на очередную регенерируемую строку памяти.
- IOCHK — контроль канала. Низкий уровень сигнала вызывает NMI CPU (разрешение и индикация в системных портах 061h, 062h).
- Reset — сигнал аппаратного сброса (активный уровень — высокий).

- BClock — синхронизация шины с частотой около 8 МГц. ПУ могут не использовать этот сигнал, работая только по управляющим сигналам записи и чтения.
- OSC — частота 14,431818 МГц, несинхронизированная с шиной (использовалась старыми дисплейными адаптерами).

Кроме логических сигналов шина имеет контакты для разводки *питания* +5, –5, +12 и –12 В.

Дополнительный разъем, расширяющий шину до 16-битной, содержит линии данных, адреса, запросов прерываний и каналов прямого доступа:

- Data[15:8] — шина данных.
- SBHE — признак наличия данных на линиях Data [15:8].
- LA[23:17] — нефиксированные сигналы адреса, требующие защелкивания по спаду сигнала BALE. Такой способ подачи адреса позволяет сократить время цикла (схемы дешифратора адреса памяти плат расширения начинают декодирование несколько раньше спада BALE).
- IRQ[10:12], IRQ[14:15] — дополнительные запросы прерываний.
- DRQ[5:7] — запросы 16-битных каналов DMA (положительным перепадом).
- DACK[5:7]# — подтверждение запросов 16-битных каналов DMA.

С *переключением разрядности данных* связаны сигналы:

- MCS16# — адресуемое устройство поддерживает 16-битные обращения к памяти.
- IOCS16# — адресуемое устройство поддерживает 16-битные обращения к портам.

К *новым управляющим сигналам* относятся следующие:

- MemWR# — запись в память в любой области до 16 Мбайт.
- MemRD# — чтение памяти в любой области до 16 Мбайт.
- 0WS# — сигнал от устройства, разрешающий системной плате укоротить текущий цикл (устранить такты ожидания).

- **MASTER#** — запрос от устройства, использующего 16-битный канал DMA на управление шиной. При получении подтверждения DACK [5:7] Bus-Master может захватить шину (не более чем на 15 мкс).

Обобщенные временные диаграммы циклов чтения или записи памяти или ввода/вывода приведены на рис. 4.5. Здесь условный сигнал CMD* изображает один из сигналов:

- **SMEMRD#, MEMRD#** — в цикле чтения памяти;
- **SMEMWR#, MEMWR#** — в цикле записи памяти;
- **IORD#** — в цикле чтения порта ввода/вывода;
- **IOWR#** — в цикле записи порта ввода/вывода.

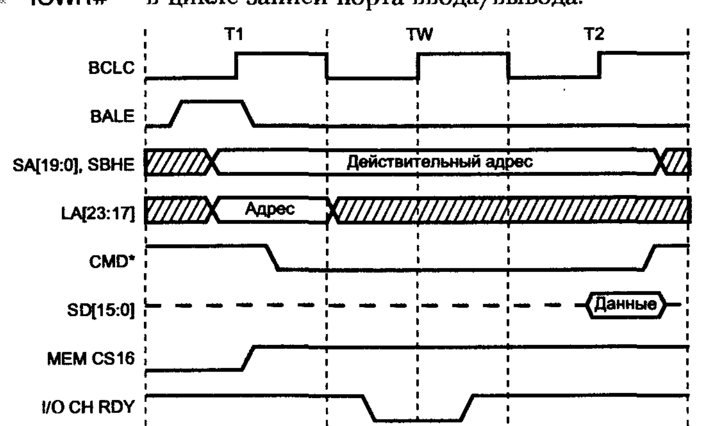


Рис. 4.5. Временные диаграммы циклов чтения или записи на шине ISA

В каждом из рассматриваемых циклов активными (с низким уровнем) могут быть только сигналы лишь из одной строки данного списка. Цикл прямого доступа к памяти, в котором это правило не соблюдается, рассмотрен в приложении А (см. рис. А.2).

По адресованному ему спаду сигнала *чтения* устройство должно выдать на шину данных содержимое адресуемой ячейки и удерживать его, пока не произойдет подъем сигнала. Во время циклов *записи* процессор выставляет действительные данные несколько позже начала (спада) сигнала записи. Уст-

ройство должно фиксировать для себя эти данные в конце цикла по подъему сигнала. Обращение к портам ввода/вывода отличается тем, что сигналы LA[32:17] не используются.

Минимальная длительность цикла определяется чипсетом и может программироваться опциями BIOS Setup заданием тактов ожидания. При этом циклы обращения к памяти короче циклов обращения к портам ввода/вывода. В шине AT для управления минимальной длительностью цикла используются также сигналы управления разрядностью передачи: если устройство поддерживает 16-битные передачи, предполагается, что оно может работать с меньшим количеством тактов ожидания. Этим объясняется, что в Setup длительности циклов ISA задаются раздельно как для памяти и ввода/вывода, так и для 8- и 16-битных операций.

Если устройство не работает с заданными циклами, оно может вводить дополнительные такты ожидания, используя сигнал IOCHRDY, но при этом недопустимо удлинять цикл более чем на 15 мкс.

Шина *EISA* позволяет устройствам работать как в режиме ISA, так и в новых режимах EISA. Из сигналов ISA используются линии адреса/данных с расширением их до 32 бит, сигналы прерываний и прямого доступа к памяти и синхронизации. Для своих циклов обмена EISA использует собственный набор управляющих сигналов с обязательной синхронизацией по сигналу BCLK. Сигналы IORD# и IOWR# при обращении к EISA-портам не используются. Вместо них применяются сигналы обращения M/IO# и W/R#. В EISA-циклах не используется сигнал SBHE; он заменен сигналами разрешения байт BE[3:0]. Вместо общего сигнала AEN каждый слот имеет собственный сигнал AENx. К *сигналам расширения шины EISA* относятся следующие:

- LA[23:16] — опережающие сигналы адреса, требующие защелкивания по спаду сигнала BALE.
- LA[31:24]# — аналогично LA [2:23], но с инверсией.
- BE[3:0]# — индикаторы действительности данных в байтах 0–3 при EISA-цикле.
- M/IO# — обращение к памяти (1) или портам (0) EISA.

- **W/R#** — запись (1) или чтение (0) EISA.
- **EX16#, EX32#** — индикаторы разрядности ресурсов.
- **SLBURST#** — адресуемое устройство поддерживает пакетный режим обмена.
- **MSBURST#** — уведомление устройства, подавшего сигнал **SLBURST**, о намерении инициировать пакетный обмен.
- **LOCK#** — обеспечение исключительного права доступа активного EISA Bus-Master к памяти или портам.
- **EXRDY** — готовность EISA-устройства; аналогично сигналу **IOCHRDCY**.
- **START#** — отмечает начало цикла EISA-доступа. Во время действия этого сигнала фиксируется адрес и определяется тип обращения.
- **CMD#** — задает временные параметры цикла; действует после сигнала **START#**.
- **MREQx#** — EISA Bus-Master слота «x» запрашивает управление шиной.
- **MAKx#** — EISA Bus-Master слота «x» получает управление шиной от арбитра.

Рис. 4.6. Временные диаграммы одиночных циклов на шине EISA

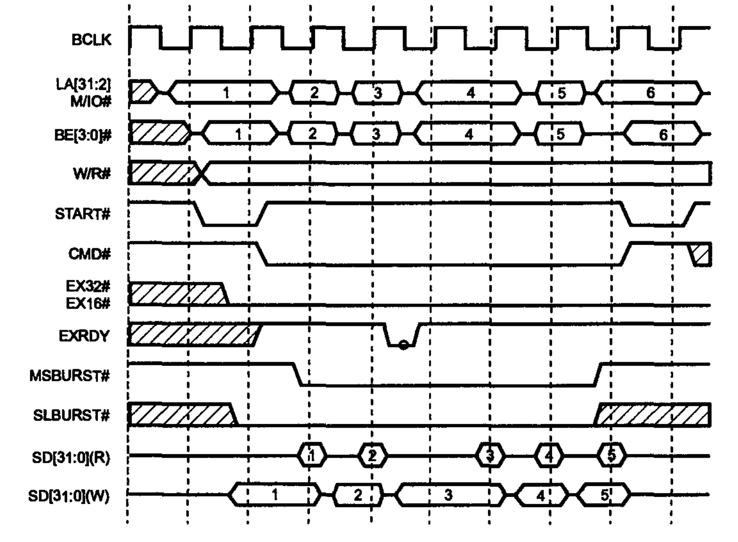


Рис. 4.7. Временные диаграммы пакетных циклов на шине EISA

Временные диаграммы одиночных и пакетных циклов обмена по шине EISA приведены на рис. 4.6 и 4.7. Обратим внимание на то, что действительные значения сигналов должны фиксироваться по положительному перепаду BCLK#, а готовность проверяется по отрицательному.

Для шин ISA ряд фирм выпускают *карты-прототипы* (Prototype Card), представляющие собой печатные платы полного или уменьшенного формата с крепежной скобой. На платах установлены обязательные интерфейсные цепи — буфер данных, дешифратор адреса и некоторые другие. Остальная часть платы свободна, и здесь разработчик может разместить макетный вариант своего устройства. Эти платы удобны для проверки нового изделия, а также для монтажа единичных экземпляров устройства, когда разработка и изготовление печатной платы нерентабельно.

На некоторых системных платах (ASUSTek) имеется небольшой разъем с загадочным названием *Media Bus*, расположенный позади разъема шины PCI одного из слотов. На этот разъем выводятся сигналы шины ISA. Он предназначен для

того, чтобы на графическом адаптере с шиной PCI можно было разместить недорогой чипсет звуковой карты, предназначенный для шины ISA. Этот разъем и такие комбинированные аудиовидеокарты широкого распространения не получили.

Обычно в PC разъемы шины расширения устанавливают на системную плату. Для компьютеров инструментального и промышленного назначения по ряду причин удобнее использовать модули, соединяемые пассивной объединяющей платой. Самый простой соответствующий конструктив заключается в использовании стандартной шины и карты ISA. Все компоненты с традиционной системной платы перенесли на карту ISA, получив одноплатный компьютер, называемый *микроPC* (mPC — microPC). На такой карте содержатся процессор, память, графический адаптер, контроллеры портов и дисковые интерфейсы, иногда дополнительные контроллеры цифрового и аналогового ввода/вывода. Для подключения к модулям расширения используют пассивную кросс-плату с обычными разъемами ISA. Если требуется более высокопроизводительный канал, применяют шину PCI. При этом на плате системного контроллера присутствуют два краевых разъема — ISA на обычном месте и PCI примерно на том месте, где располагался разъем VLB. Кросс-плата для таких систем становится неоднородной — часть ее слотов имеет разъемы PCI, другая часть — ISA, место для системного контроллера оборудовано обоими разъемами.

Логически эквивалентной ISA является шина *PC-104*, предназначенная для построения встраиваемых контроллеров. 104 — число контактов коннектора, на который выводятся сигналы шины ISA. От ISA шина PC-104 отличается только типом коннектора и нагрузочными характеристиками линий. Основой контроллера является mPC с разъемом PC-104 (розетка). Если требуется подключение платы расширения, она своей вилкой PC-104 вставляется в плату контроллера. Кроме вилки на плате расширения имеется розетка PC-104 (коннектор двухсторонний), так что можно собрать «бутерброд» из нескольких плат. Если плат более трех, то сверху «бутерброда» устанавливают терминатор. Для фиксации плат стандартизовано расположение крепежных отверстий. Платы скрепляются несущими стойками (длинными винтами со втулками).

4.2. Шина MCA

MCA (MicroChannel Architecture) — микроканальная архитектура — была введена фирмой IBM для своих компьютеров PS/2, начиная с модели 50. Шина MCA абсолютно несовместима с ISA/EISA. Состав управляющих сигналов, протокол и архитектура ориентированы на асинхронное функционирование шины и процессора, что снимает проблемы согласования скоростей процессора и ПУ. Архитектура позволяет эффективно и автоматически конфигурировать все устройства программным путем (в картах MCA нет ни одного переключателя).

Слоты MCA имеют контакты с шагом 0,05 дюйма. Число слотов на системной плате зависит от модели компьютера.

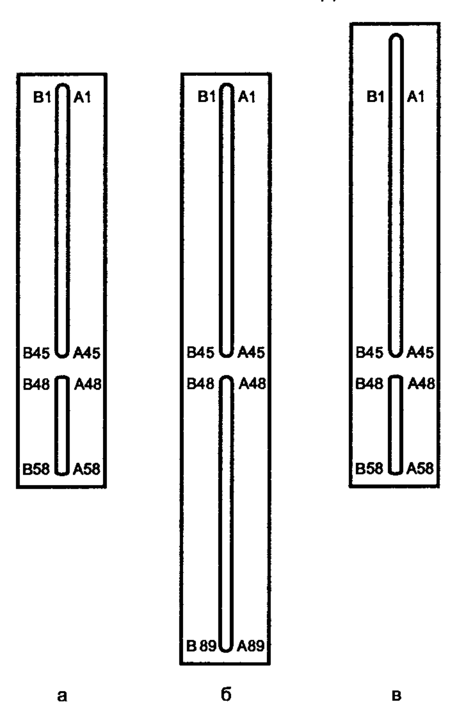


Рис. 4.8. Слоты MCA: *а* — 16-битный, *б* — 32-битный, *в* — 16-битный с расширением

- *16-битный слот* (рис. 4.8а) использует контакты A1/B1–A45/B45 для 8-битных операций и A48/B48–A58/B58 для 16-битных. Пропущенные номера 46, 47 — ключ для всех слотов.
- *32-битный слот* (рис. 4.8б) имеет удлиненную вторую секцию с контактами A48/B48–A89/B89.

Перед контактами A1/B1 возможно одно из двух *расширений* за счет удлинения первой секции (рис. 4.8в):

- AM1/BM1–AM4/BM4 — для плат расширения памяти;
- AV1/BV1–AV10/BV10 — для плат видеоадаптеров. Позволяет использовать общие цепи встроенного в системную плату VGA-адаптера, теоретически удешевляя расширение видеосистемы. По составу сигналов видеорасширение близко к разъему VGA Auxiliary Video Connector или VESA Feature Connector.

При всей прогрессивности архитектуры MCA не пользуется популярностью, в частности, из-за полной несовместимости с ISA. MCA находит применение в мощных файл-серверах, где требуется высоконадежный производительный ввод/вывод.

4.3. Локальная шина VLB

Шины ввода/вывода ISA, MCA, EISA имеют низкую производительность, обусловленную их ролью в структуре PC. Современные приложения требуют существенного повышения пропускной способности шины. Одним из решений проблемы стало применение локальной шины процессора 486. Шину процессора использовали как место подключения периферии (контроллера дисков и графического адаптера).

VLB (VESA Local Bus) — стандартизованная 32-битная локальная шина, практически представляющая собой сигналы системной шины процессора 486, выведенные на дополнительные разъемы системной платы. Шина ориентирована на процессор 486, хотя возможно ее использование с 386. Для

процессоров Pentium была принята спецификация 2.0, в которой разрядность шины данных увеличена до 64, но она распространения не получила.

Конструктивно VLB-слот аналогичен 16-битному MCA-слоту, но является расширением системного слота шины ISA-16, EISA или MCA, располагаясь позади него вблизи от процессора. Из-за ограниченной нагрузочной способности шины процессора больше трех слотов VLB на системной плате не устанавливают. Максимальная тактовая частота шины — 66 МГц, хотя шина работает надежнее на частоте 33 МГц. Декларируется пиковая пропускная способность 132 Мбайт/с (33 МГц×4 байта), но она достигается только при пакетной передаче. Реально в пакетном цикле передача 4×4 = 16 байт данных требует 5 тактов шины, так что даже в нем пропускная способность составляет 105,6 Мбайт/с, а в обычном режиме (по такту на фазу адреса и на фазу данных) — 66 Мбайт/с. Жесткие требования к временным характеристикам процессорной шины при большой нагрузке (со стороны микросхем внешнего кэша) могут привести к неустойчивой работе: все три VLB-слота могут использоваться только на частоте 40 МГц. При нагруженной системной плате на 50 МГц может работать только один слот. Шина допускает применение активных (Bus-Master) адаптеров, но арбитраж запросов возлагается на сами адаптеры. Разрешена установка не более двух Bus-Master-адаптеров, один из которых устанавливается в «Master»-слот.

Шину VLB использовали для подключения графического адаптера и контроллера дисков. Встречаются системные платы, которые имеют встроенный графический и дисковый адаптеры с шиной VLB, но самих слотов VLB нет. Это означает, что на плате установлены микросхемы указанных адаптеров для шины VLB. Такая неявная шина по производительности не уступает обычной, а с точки зрения надежности и совместимости она даже лучше.

Назначение контактов шины VLB представлено в табл. 4.4. В названиях сигналов можно узнать сигналы процессора 486.

Таблица 4.4. Разъем шины VLB

Ряд B		№	Ряд A	
32/64-битный обмен	64-битный обмен		32/64-битный обмен	64-битный обмен
Data 0	–	1	Data 1	–
Data 2	–	2	Data 3	–
Data 4	–	3	GND	–
Data 6	–	4	Data 5	–
Data 8	–	5	Data 7	–
GND	–	6	Data 9	–
Data 10	–	7	Data 11	–
Data 12	–	8	Data 13	–
VCC	–	9	Data 15	–
Data 14	–	10	GND	–
Data 16	–	11	Data 17	–
Data 18	–	12	VCC	–
Data 20	–	13	Data 19	–
GND	–	14	Data 21	–
Data 22	–	15	Data 23	–
Data 24	–	16	Data 25	–
Data 26	–	17	GND	–
Data 28	–	18	Data 27	–
Data 30	–	19	Data 29	–
VCC	–	20	Data 31	–
Addr 31	Data 63	21	Addr 30	Data 62
GND	–	22	Addr 28	Data 60
Addr 29	Data 61	23	Addr 26	Data 58
Addr 27	Data 59	24	GND	–
Addr 25	Data 57	25	Addr 24	Data 56
Addr 23	Data 55	26	Addr 22	Data 54
Addr 21	Data 53	27	VCC	–

Ряд B		№	Ряд A	
32/64-битный обмен	64-битный обмен		32/64-битный обмен	64-битный обмен
Addr 19	Data 51	28	Addr 20	Data 52
GND	–	29	Addr 18	Data 50
Addr 17	Data 49	30	Addr 16	Data 48
Addr 15	Data 47	31	Addr 14	Data 46
VCC	–	32	Addr 12	Data 44
Addr 13	Data 45	33	Addr 10	Data 42
Addr 11	Data 43	34	Addr 8	Data 40
Addr 9	Data 41	35	GND	–
Addr 7	Data 39	36	Addr 6	Data 38
Addr 5	Data 37	37	Addr 4	Data 36
GND	–	38	Write Back	–
Addr 3	Data 35	39	Byte Enable 0	Byte Enable 4
Addr 2	Data 34	40	VCC	–
Unused	LBS64#	41	Byte Enable 1	Byte Enable 5
Reset	–	42	Byte Enable 2	Byte Enable 6
Data-Code Status	–	43	GND	–
Mem-I/O Status	Data 33	44	Byte Enable 3	Byte Enable 7
Wr-Rd Status	Data 32	45	Address Data Strobe	–
ключ	–	46	ключ	–
ключ	–	47	ключ	–
Ready Return	–	48	Local Ready	–
GND	–	49	Local Device	–
IRQ 9	–	50	Local Request	–
Burst Ready	–	51	GND	–
Burst Last	–	52	Local Bus Grant	–

Ряд B		№	Ряд A	
32/64-битный обмен	64-битный обмен		32/64-битный обмен	64-битный обмен
ID 0	–	53	VCC	–
ID 1	–	54	ID 2	–
GND	–	55	ID 3	–
Local Clock	–	56	ID 4	ACK64#
VCC	–	57	Unused	–
Local Bus Size 16	–	58	Loc/Ext Address Data Strobe	–

4.4. Шина PCI

PCI (Peripheral Component Interconnect) local bus — шина соединения периферийных компонентов. Будучи локальной, эта шина занимает особое место в современной архитектуре PC, являясь мостом (mezzanine bus) между системной шиной процессора и шиной ввода/вывода ISA/EISA или MCA. Шина PCI разрабатывалась в расчете на Pentium-системы, но хорошо сочетается и с процессорами 486. Является четко стандартизованной высокопроизводительной и надежной шиной расширения. В настоящее время действует спецификация PCI 2.1. При частоте шины 33 МГц скорость теоретически достигает 132/264 Мбайт/с для 32/64 бит; при частоте синхронизации 66 МГц — в два раза выше (версия 2.1 допускает частоту до 66 МГц при согласии всех устройств на шине).

Шина является *синхронной* — фиксация всех сигналов выполняется по положительному перепаду (фронту) сигнала CLK. В каждой транзакции (обмене по шине) участвуют два устройства — *инициатор обмена* (Initiator или Master, инициирующее устройство, ИУ) и *целевое устройство* (Target или Slave, ЦУ).

Шина PCI все транзакции трактует как пакетные: каждая транзакция начинается фазой адреса, за которой может сле-

довать одна или несколько фаз данных. Для адреса и данных используются общие мультиплексированные линии AD. Четыре мультиплексированные линии C/BE[3:0] используются для кодирования команд в фазе адреса и разрешения байт в фазе данных. В начале транзакции ИУ активизирует сигнал FRAME#, по шине AD передает целевой адрес, а по линиям C/BE# — информацию о типе транзакции (команде). Адресованное ЦУ отзывается сигналом DEVSEL#, после чего ИУ может указать на свою готовность к обмену данными сигналом IRDY#. Когда к обмену данными будет готово и ЦУ, оно установит сигнал TRDY#. Данные по шине AD могут передаваться только при одновременном наличии сигналов IRDY# и TRDY#. С помощью этих сигналов ИУ и ЦУ согласуют свои скорости, вводя такты ожидания. На рис. 4.9 приведена временная диаграмма обмена, в которой и ИУ, и ЦУ вводят такты ожидания. Если бы они оба ввели сигналы готовности в конце фазы адреса и не снимали их до конца обмена, то в каждом такте после фазы адреса передавались бы по 32 бита данных, что обеспечило бы выход на предельную производительность обмена.

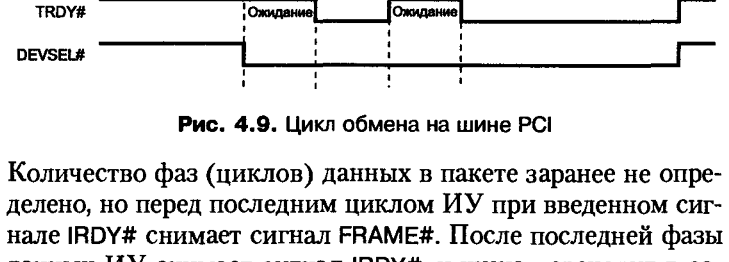


Рис. 4.9. Цикл обмена на шине PCI

Количество фаз (циклов) данных в пакете заранее не определено, но перед последним циклом ИУ при введенном сигнале IRDY# снимает сигнал FRAME#. После последней фазы данных ИУ снимает сигнал IRDY#, и шина переходит в состояние покоя (*PCI Idle*) — оба сигнала FRAME# и IRDY# находятся в пассивном состоянии. Максимальное количество

ется индивидуальным сигналом IDSEL#, конфигурационные регистры выбираются двойными словами по линиям AD[7:2], при этом AD[1:0]=00. Сигнал выборки IDSEL# воспринимается устройством только в фазе адреса, поэтому обычно в качестве него используют старшие биты адреса с позиционным кодированием адреса устройства.

Команды шины PCI определяются значениями бит C/BE# в фазе адреса в соответствии с табл. 4.5.

Таблица 4.5. Декодирование команд шины PCI

C/BE[3:0]	Тип команды
0000	*Interrupt Acknowledge* — подтверждение прерывания
0001	*Special Cycle* — специальный цикл
0010	*I/O Read* — чтение порта ввода/вывода
0011	*I/O Write* — запись в порт ввода/вывода
0100	Зарезервировано
0101	Зарезервировано
0110	*Memory Read* — чтение памяти
0111	*Memory Write* — запись в память
1000	Зарезервировано
1001	Зарезервировано
1010	*Configuration Read* — конфигурационное считывание
1011	*Configuration Write* — конфигурационная запись
1100	*Multiple Memory Read* — множественное чтение памяти
1101	*Dual Address Cycle* — двухадресный цикл
1110	*Memory Read Line* — чтение строк памяти
1111	*Memory Write and Invalidate* — запись с инвалидацией

В команде *подтверждение прерывания* контроллер прерываний передает вектор прерывания по шине AD.

Специальный цикл декодируется содержимым линий AD[15:0] и используется для указания на отключение (*Shutdown*), останов (*Halt*) процессора или специфические функции процессора, связанные с кэшем и трассировкой. Этим состояниям соответствуют коды 0000, 0001 и 0002; коды 0003-FFFFh зарезервированы.

В командах *чтения и записи ввода/вывода* линии AD содержат адрес байта, причем декодированию подлежат и биты AD0 и AD1 (несмотря на то, что имеются сигналы BEx#). Порты PCI могут быть 8-, 16- или 32-битными. Для адресации портов на шине PCI доступны все 32 бита адреса, но процессоры x86 могут использовать только младшие 16 бит. Порты с адресами 0CF8 и 0CFC зарезервированы под регистры адреса и данных для доступа к конфигурационному пространству. Обращение к порту данных приведет к генерации шинного цикла конфигурационного чтения или записи по предварительно записанному адресу.

В командах *чтения и записи памяти* шина AD содержит адреса двойных слов, и линии AD0, AD1 не должны декодироваться — на конкретные байты указывают сигналы C/BE[3:0]#.

Команды *конфигурационного чтения и записи* адресуются к конфигурационному пространству и обеспечивают доступ к 256-байтным структурам. Обращение производится двойными словами. Структура содержит идентификатор устройства и производителя, состояние и команду, информацию об используемых ресурсах и ограничения на использование шины.

Множественное чтение памяти используется для транзакций, пересекающих границы строк кэш-памяти.

Двухадресный цикл применяется, когда физическая шина имеет 32 бита адреса, а требуется передача с 64-битной адресацией. В этом случае младшие 32 бита адреса передаются в цикле данного типа, а за ним следует обычный цикл, определяющий тип обмена и несущий старшие 32 бита адреса. Шина PCI допускает 64-битную адресацию даже для портов ввода/вывода (для x86 это бесполезно, но PCI существует и на других платформах).

Чтение строк памяти применяется, когда в транзакции планируется более двух 32-битных передач (обычно это чтение до конца строки кэша).

Запись с инвалидацией применяется к целым строкам кэша и позволяет оптимизировать циклы обратной записи «грязных» строк кэша.

Выделение специальных циклов множественного чтения, чтения строк и записи с инвалидацией позволяет контроллеру памяти предпринять определенные меры для оптимизации данных передач.

Слоты PCI с шагом 0,05 дюйма расположены несколько дальше от задней панели, чем ISA/EISA или MCA. Компоненты карт PCI расположены на левой поверхности плат. По этой причине крайний PCI-слот обычно разделяет использование посадочного места адаптера с соседним ISA-слотом (Shared slot). Шина имеет версии с питанием *5 В*, *3,3 В* и *универсальную* (с переключением линий +V I/O с 5 В на 3,3 В). Ключами являются пропущенные ряды контактов 12, 13 и 50, 51. Для слота на 5 В ключ расположен на месте контактов 50, 51; для 3 В — 12, 13; для универсального — два ключа: 12, 13 и 50, 51. Ключи не позволяют установить карту в слот с неподходящим напряжением питания. 32-битный слот заканчивается контактами A62/B62, 64-битный — A94/B94. На рис. 4.10 изображена 32-битная плата максимального размера (Long Card), длина короткой платы (Short Card) — 6,875". Плата может иметь обрамление (скобки), стандартное для конструктива ISA или MCA. Назначение выводов универсального разъема приведено в табл. 4.6, назначение сигналов — в табл. 4.7.

Рис. 4.10. Карта расширения для шины PCI

На одной шине PCI может быть не более четырех устройств (следовательно, и слотов). Для подключения к другим шинам применяются специальные аппаратные средства — *мос-*

ты PCI (PCI Bridge). *Главный мост* (Host Bridge) используется для подключения PCI к системной шине. *Одноранговый мост* (Peer-to-Peer Bridge) используется для соединения двух шин PCI. Несколько шин PCI применяются в серверах — это позволяет увеличить число подключаемых устройств. Для подключения шин ISA/EISA используются специальные мосты, входящие в чипсеты системных плат. Каждый мост программируется — ему указываются диапазоны адресов пространств памяти и ввода/вывода, отведенные устройствам его шин. Если адрес ЦУ текущей транзакции на одной шине (стороне) моста относится к шине противоположной стороны, мост перенаправляет транзакцию на соответствующую шину и выполняет действия по согласованию протоколов шин. Таким образом, совокупность мостов PCI выполняет маршрутизацию (*routing*) обращений по связанным шинам. Считается, что устройство с конкретным адресом может присутствовать только на одной из шин, а на какой именно, «знают» запрограммированные мосты. Решать задачу маршрутизации призван также сигнал **DEVSEL#**. Обращения, не востребованные абонентами PCI, обычно перенаправляются на шину ISA/EISA.

Таблица 4.6. Разъемы шины PCI

Ряд B	N#	Ряд A	Ряд B	N#	Ряд A
–12 В	1	TRST#	PRSNT2#	11	Reserved
TCK	2	+12 В	GND/Ключ 3,3 В	12	GND/Ключ 3,3 В
GND	3	TMS	GND/Ключ 3,3 В	13	GND/Ключ 3,3 В
TDO	4	TDI	Reserved	14	Reserved
+5 В	5	+5 В	GND	15	RST#
+5 В	6	INTRA#	Clock	16	+V I/O
INTRB#	7	INTRC#	GND	17	GNT#
INTRD#	8	+5 В	REQ#	18	GND
PRSNT1#	9	Reserved	+V I/O	19	Reserved
Reserved	10	+V I/O	AD 31	20	AD 30

Таблица 4.8 (продолжение)

Ряд В	N#	Ряд А	Ряд В	N#	Ряд А
AD 29	21	+3,3 В	GND/Ключ 5 В	50	GND/Ключ 5 В
GND	22	AD 28	GND/Ключ 5 В	51	GND/Ключ 5 В
AD 27	23	AD 26	AD 8	52	C/BE0#
AD 25	24	GND	AD 7	53	+3,3 В
+3,3 В	25	AD 24	+3,3 В	54	AD 6
C/BE3#	26	IDSEL#	AD 5	55	AD 4
AD 23	27	+3,3 В	AD 3	56	GND
GND	28	AD 22	GND	57	AD 2
AD 21	29	AD 20	AD 1	58	AD 0
AD 19	30	GND	+V I/O	59	+V I/O
+3.3 В	31	AD 18	ACK64#	60	REQ64#
AD 17	32	AD 16	+5 В	61	+5 В
C/BE2#	33	+3,3 В	+5 В	62	+5 В
GND	34	FRAME#	Конец 32-битного разъема		
IRDY#	35	GND	Reserved	63	GND
+3,3 В	36	TRDY#	GND	64	C/BE7#
DEVSEL#	37	GND	C/BE6#	65	C/BE5#
GND	38	STOP#	C/BE4#	66	+V I/O
LOCK#	39	+3,3 В	GND	67	PAR64
PERR#	40	SDONE#	AD 63	68	AD 62
+3,3 В	41	SDOFF#	AD 61	69	GND
SERR#	42	GND	+V I/O	70	AD 60
+3,3 В	43	PAR	AD 59	71	AD 58
C/BE1#	44	AD 15	AD 57	72	GND
AD 14	45	+3,3 В	GND	73	AD 56
GND	46	AD 13	AD 55	74	AD 54
AD 12	47	AD 11	AD 53	75	+V I/O
AD 10	48	GND	GND	76	AD 52
GND/M66EN[1]	49	AD 9	AD 51	77	AD 50

Ряд B	N#	Ряд A	Ряд B	N#	Ряд A
AD 49	78	GND	AD 37	87	GND
+V I/O	79	AD 48	+V I/O	88	AD 36
AD 47	80	AD 46	AD 35	89	AD 34
AD 45	81	GND	AD 33	90	GND
GND	82	AD 44	GND	91	AD 32
AD 43	83	AD 42	Reserved	92	Reserved
AD 41	84	+V I/O	Reserved	93	GND
GND	85	AD 40	GND	94	Reserved
AD 39	86	AD 38	Конец 64-битного разъема		

[1] Сигнал M66EN определен в PCI 2.1.

Таблица 4.7. Сигналы шины PCI

Сигнал	Назначение
AD[31:0]	*Address/Data* – мультиплексированная шина адреса/данных. В начале транзакции передается адрес, в последующих тактах – данные
C/BE[3:0]#	*Command/Byte Enable* – команда/разрешение обращения к байтам. Команда, определяющая тип очередного цикла шины, задается четырехбитным кодом в фазе адреса
FRAME#	Кадр. Введением сигнала отмечается начало транзакции (фаза адреса), снятие сигнала указывает на то, что последующий цикл передачи данных является последним в транзакции
DEVSEL#	*Device Select* – устройство выбрано (ответ ЦУ на адресованную ему транзакцию)
IRDY#	*Initiator Ready* – готовность ИУ к обмену данными
TRDY#	*Target Ready* – готовность ЦУ к обмену данными
STOP#	Запрос ЦУ к ИУ на останов текущей транзакции
LOCK#	Используется для установки, обслуживания и освобождения захвата ресурса PCI
REQ#	*Request* – запрос от PCI-мастера на захват шины
GNT#	*Grant* – предоставление мастеру управления шиной
PAR	*Parity* – общий бит паритета для линий AD[31:0] и C/BE[3:0]

Сигнал	Назначение
PERR#	*ParityError* — сигнал об ошибке паритета (от устройства, ее обнаружившего)
PRSNT[1,2]#	*Present* — индикаторы присутствия платы, кодирующие запрос потребляемой мощности
RST#	*Reset* — сброс всех регистров в начальное состояние
IDSEL#	*Initialization Device Select* — выбор устройства в циклах конфигурационного считывания и записи
SERR	*System Error* — системная ошибка. Активизируется любым устройством PCI и вызывает NMI
REQ64#	*Request 64 bit* — запрос на 64-битный обмен
ACK64#	Подтверждение 64-битного обмена
INTRA# INTRB# INTRC# INTRD#	*Interrupt A, B, C, D* — линии запросов прерывания. Циклически сдвигаются в слотах и направляются на доступные линии IRQ. Запрос по низкому уровню допускает разделяемое использование линий
CLK	*Clock* — тактовая частота шины. Должна быть в пределах 20–33 МГц, в PCI 2.1 — до 66 МГц
M66EN	*66MHzEnable* — разрешение частоты синхронизации до 66 МГц
SDONE	*Snoop Done* — сигнал завершенности цикла слежения для текущей транзакции. Низкий уровень указывает на незавершенность цикла слежения за когерентностью памяти и кэша. Необязательный сигнал, используется только устройствами шины с кэшируемой памятью
SBO#	*Snoop Backoff* — попадание текущего обращения к памяти абонента шины в модифицированную строку кэша. Необязательный сигнал, используется только абонентами шины с кэшируемой памятью при алгоритме обратной записи
TCK	*Test Clock* — синхронизация тестового интерфейса JTAG
TDI	*Test Data Input* — входные данные тестового интерфейса JTAG
TDO	*Test Data Output* — выходные данные тестового интерфейса JTAG
TMS	*Test Mode Select* — выбор режима для тестового интерфейса JTAG
TRST	*Test Logic Reset* — сброс тестовой логики

Одной из особенностей шины PCI является возможность обмена данными между процессором и памятью одновременно с обменом между другими устройствами PCI — *Concurrent PCI Transferring*. Эта возможность реализуется не всеми чипсетами (в описаниях она всегда специально подчеркивается), а обычными устройствами (видеокарты и контроллеры дисков) используется редко.

С мостами PCI/ISA связано понятие *VGA Palette Snooping*, которое обеспечивает исключение из правила однозначной маршрутизации обращений. Графическая карта в компьютере с шиной PCI обычно устанавливается на шину PCI. На VGA-карте имеются регистры палитр (Palette Registers), приписанные к пространству ввода/вывода. Если графическая система содержит еще и карту смешивания сигналов графического адаптера с сигналом «живого видео», перехватывая двоичную информацию о цвете текущего пиксела по шине *VESA Feature Connector* (снимаемую до регистра палитр), цветовая гамма будет определяться регистром палитр, размещенным на этой дополнительной карте. Возникает ситуация, когда операция записи в регистр палитр должна отрабатываться одновременно и в графическом адаптере (на шине PCI), и в карте видеорасширения, которая часто размещается на шине ISA. В BIOS Setup может присутствовать опция *PCI VGA Palette Snoop*. При ее разрешении запись в порты ввода/вывода по адресу регистра палитр будет вызывать транзакцию как на шине PCI, так и на шине ISA, чтение же по этим адресам будет выполняться только PCI. Реализация может возлагаться на графическую карту PCI. Для этого она во время записи в регистр палитр фиксирует данные, но сигналы квитирования DEVSEL# и TRDY# не вырабатывает, в результате чего мост распространяет этот неопознанный запрос на шину ISA. В других реализациях мосту явно указывают на данное исключение, и он сам распространяет запись в регистры палитр на шину ISA.

Автоконфигурирование устройств (выбор адресов и прерываний) поддерживается средствами BIOS и ориентировано на технологию Plug and Play. Стандарт PCI определяет для каждого слота конфигурационное пространство размером до 256 8-битных регистров, не приписанных ни к простран-

ству памяти, ни к пространству ввода/вывода. Доступ к ним осуществляется по специальным циклам шины Configuration Read и Configuration Write, вырабатываемым контроллером при обращении процессора к регистрам контроллера шины PCI, расположенным в его пространстве ввода/вывода. После аппаратного сброса (или по включении питания) устройства PCI не отвечают на обращения к пространству памяти и ввода/вывода, они доступны только для операций конфигурационного считывания и записи. В этих операциях устройства выбираются по индивидуальным сигналам IDSEL# и сообщают о потребностях в ресурсах и возможных диапазонах памяти. После распределения ресурсов, выполняемого программой конфигурирования (во время POST), в устройства записываются параметры конфигурирования. Только после этого к устройствам становится возможным доступ по командам обращения к памяти и портам ввода/вывода.

Для ПЗУ расширения BIOS, установленных на картах PCI, принят стандарт, несколько отличающийся от традиционных дополнительных модулей ROM BIOS. Поскольку шина PCI используется не только в PC, в ПЗУ карты может храниться несколько модулей. Тип платформы (процессора) указывается в заголовке модуля, и при инициализации BIOS активизируется нужный. Такой механизм позволяет, например, один и тот же графический адаптер устанавливать и в IBM PC, и в Power PC.

В состав шины PCI введены сигналы для тестирования адаптеров по интерфейсу *JTAG*. На системной плате эти сигналы задействованы не всегда, но они могут и организовывать логическую цепочку тестируемых адаптеров.

Некоторые фирмы выпускают для PCI карты-прототипы, но доукомплектовать их адаптером собственной разработки сложнее, чем карту ISA. Здесь сказываются сложные протоколы и высокие частоты.

Слот PCI достаточен для подключения адаптера (в отличие от VLB), на системной плате он может сосуществовать с любой из шин ввода/вывода и даже с VLB. На некоторых системных платах позади одного из слотов PCI имеется разъем Media Bus, на который выводятся сигналы ISA. Он

используется для размещения на графическом адаптере PCI звукового чипсета, предназначенного для шины ISA.

Для *устройств промышленного назначения* в начале 1995 г. был принят стандарт *Compact PCI*. Шина Compact PCI (*cPCI*) разрабатывалась на основе спецификации PCI 2.1. Эта шина отличается большим количеством поддерживаемых слотов: 8 против 4. Появились новые 4 пары сигналов запросов и предоставления управления шиной. Шина поддерживает 32-битный и 64-битный обмен (с индивидуальным разрешением байт). При частоте шины 33 МГц максимальная пропускная способность составляет 133 Мбайт/с для 32 бит и 266 Мбайт/с для 64 бит (в середине пакетного цикла). Возможна работа на частоте 66 МГц. Шина поддерживает спецификацию PnP. Кроме того, в шине возможно применение *географической адресации*, при этом адрес модуля (на который он отзывается при программном обращении) определяется его положением в каркасе. Для этого на коннекторе J1 имеются контакты GA0...GA4, коммутацией которых на «землю» для каждого слота задается его двоичный адрес. Географическая адресация позволяет переставлять однотипные модули, не заботясь о конфигурировании их адресов (хорошая альтернатива системе PnP — модуль «встанет» всегда в одни и те же адреса, которые без физического вмешательства больше не изменятся). Конструктивно платы Compact PCI представляют собой еврокарты высотой 3U (100×160 мм) с одним коннектором (J1) или 6U (233×160 мм) с двумя коннекторами (J1 и J2). Коннекторы — 7-рядные штырьковые разъемы с шагом 2 мм между контактами, на кросс-плате — вилка, на модулях — розетки. Контакты коннекторов имеют разную длину: более длинные контакты цепей питания при установке модуля соединяются раньше, а при вынимании разъединяются позже, чем сигнальные. Такое решение позволяет производить «горячую» замену модулей. Собственно шина использует только один коннектор (J1), причем в 32-битном варианте не полностью — часть контактов может задействовать пользователь. 64-битная шина использует коннектор полностью. Одно посадочное место на кросс-плате резервируется под контроллер шины, на который возлагаются функции арбитража и синхронизации. На его

коннекторе шиной используется большее число контактов, чем на остальных. У больших плат коннектор J2 предоставляется пользователю, а между коннекторами J1 и J2 может устанавливаться 95-контактный коннектор J3. Конструкция коннекторов позволяет для J2 применять специфические модификации (например, с разделяющим экраном и механическими ключами). В шине предусматривается наличие независимых источников питания +5 В, +3,3 В и ±12 В.

На базе шины Compact PCI фирмой National Instruments разработана *спецификация PXI* (PCI eXtensions for Instrumentation — расширение PCI для инструментальных систем) в тех же конструктивах. В шине PXI часть контактов, определенных в Compact PCI как свободно используемые, предназначается для дополнительных шин. Шина *Trigger Bus* (8 линий) звездообразно соединяет слот своего контроллера (первый после системного контроллера PCI) с остальными слотами. Шина позволяет осуществлять синхронизацию разных модулей, что зачастую требуется в измерительных системах. Для прецизионной синхронизации имеется сигнал опорной частоты 10 МГц PXI_CLK, который звездообразно (с одинаковыми задержками распространения сигнала) разводится по слотам. В PXI определены локальные шины, предназначенные для связи соседних пар слотов. Каждая локальная шина имеет 13 линий, которые могут использоваться как для цифровых, так и для аналоговых (до 48 В) сигналов. Локальные шины объединяют смежные слоты попарно (исключая слот системного контроллера), образуя цепочку. Кроме механических и электрических характеристик PXI определяет ПО модулей: основной ОС считается Windows NT/95, и модули должны поставляться с соответствующими драйверами. Это экономит время, необходимое для системной интеграции. Модули PXI совместимы с шиной Compact PCI, и модули Compact PCI — с шиной PXI. Однако все преимущества спецификации реализуются только при установке модулей PXI в шину PXI.

4.5. Магистральный интерфейс AGP

В настоящее время самой быстрой универсальной шиной расширения является PCI, имеющая при тактовой частоте

66 МГц и разрядности 32 бит пиковую пропускную способность 264 Мбайт/с. Одним из главных потребителей пропускной способности шины является графический адаптер. По мере увеличения разрешения и глубины цвета требования к пропускной способности шины, связывающей дисплейный адаптер с памятью и центральным процессором компьютера, повышаются. Одно из решений состоит в уменьшении потока графических данных, передаваемых по шине. Для этого графические платы снабжают акселераторами и увеличивают объем видеопамяти, которой пользуется акселератор при выполнении построений. В результате поток данных в основном циркулирует внутри графической карты, слабо нагружая внешнюю шину. Однако при трехмерных построениях акселератору становится тесно в ограниченном объеме видеопамяти, и его поток данных снова выплескивается на внешнюю шину.

Фирма Intel на базе шины PCI 2.1 разработала новый стандарт подключения графических адаптеров — *AGP* (Accelerated Graphic Port — ускоренный графический порт). Этот порт представляет собой 32-разрядную шину с тактовой частотой 66 МГц, по составу сигналов (табл. 4.8) напоминающую шину PCI. Чипсет системной платы связывает AGP с памятью и системной шиной процессора, не пересекаясь с «узким местом» — шиной PCI. «Ускоренность» порта обеспечивается следующими факторами:

- конвейеризацией обращений к памяти;
- сдвоенной передачей данных;
- демультиплексированием шин адреса и данных.

Конвейеризацию обращений к памяти иллюстрирует рис. 4.11, где сравниваются обращения к памяти PCI и AGP. В PCI во время реакции памяти на запрос шина простаивает. Конвейерный доступ AGP позволяет в это время передавать следующие запросы, а потом получить поток ответов. AGP предусматривает постановку в очередь до 256 запросов, но при конфигурировании по PnP реальные возможности конкретной системы уточняются (возможности контроллера памяти ограничены). AGP поддерживает две пары очередей для операций записи и чтения из памяти с

высоким и низким приоритетами. В передачу данных любого запроса может вмешаться следующий запрос, в том числе запрос в режиме PCI.

Таблица 4.8. Сигналы порта AGP

Ряд A	№	Ряд B	Ряд A	№	Ряд B
Spare	1	12V	Vddq3.3	34	Vddq3.3
5.0V	2	Spare	AD21	35	AD22
5.0V	3	Reserved	AD19	36	AD20
USB+	4	USB-	GND	37	GND
GND	5	GND	AD17	38	AD18
INTB#	6	INTA#	C/BE2#	39	AD16
CLK	7	RST#	Vddq3.3	40	Vddq3.3
REQ#	8	GNT#	IRDY#	41	FRAME#
VCC3.3	9	VCC3.3		42	
ST0	10	ST1	GND	43	GND
ST2	11	Reserved		44	
RBF#	12	PIPE#	VCC3.3	45	VCC3.3
GND	13	GND	DEVSEL#	46	TRDY#
Spare	14	Spare	Vddq3.3	47	STOP#
SBA0	15	SBA1	PERR#	48	Spare
VCC3.3	16	VCC3.3	GND	49	GND
SBA2	17	SBA3	SERR#	50	PAR
SB_STB	18	Reserved	C/BE1#	51	AD15
GND	19	GND	Vddq3.3	52	Vddq3.3
SBA4	20	SBA5	AD14	53	AD13
SBA6	21	SBA7	AD12	54	AD11
KEY	22	KEY	GND	55	GND
KEY	23	KEY	AD10	56	AD9
KEY	24	KEY	AD8	57	C/BE0#
KEY	25	KEY	Vddq3.3	58	Vddq3.3
AD31	26	AD30	AD_STB0	59	Reserved
AD29	27	AD28	AD7	60	AD6
VCC3.3	28	VCC3.3	GND	61	GND
AD27	29	AD26	AD5	62	AD4
AD25	30	AD24	AD3	63	AD2
GND	31	GND	Vddq3.3	64	Vddq3.3
AD_STB1	32	Reserved	AD1	65	AD0
AD23	33	C/BE3#	SMB0	66	SMB1

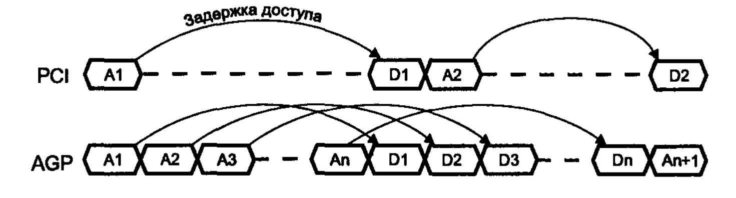


Рис. 4.11. Циклы обращения к памяти PCI и AGP

Сдвоенная передача данных обеспечивает при частоте 66 МГц пропускную способность до 532 Мбайт/с, что для 32-битной шины неожиданно. В AGP кроме «классического» режима, называемого теперь «*x1*», в котором за один такт синхронизации передается один 4-байтный блок данных, есть режим «*x2*», когда блоки данных передаются как по фронту, так и по спаду сигнала синхронизации (как в ATA Ultra DMA). Управление передачей в таком режиме названо *SideBand Control* (приставка SB к имени сигнала). Заказать режим *x2* может только графическая карта, если она его поддерживает. При переходе на тактовую частоту 100 МГц пропускная способность повышается до 800 Мбайт/с.

Демультиплексирование (разделение) *шины адреса и данных* сделано несколько необычным образом. Демультиплексирование подразумевает наличие двух полноразрядных шин — адреса и данных. Однако такой вариант дорог. Поэтому шину адреса в демультиплексированном режиме AGP представляют 8 линий SBA (*SideBand Address*), по которым за три такта синхронизации передаются четыре байта адреса, длина запроса (1 байт) и команда (1 байт). За каждый такт передаются по два байта — один по фронту, другой по спаду тактового сигнала. Поддержка демультиплексированной адресации не является обязательной для карты AGP. Требуется также поддержка хост-контроллера. Альтернативным способом подачи адреса является обычный — по мультиплексированной шине AD.

AGP может реализовать всю пропускную способность 64-битной системы памяти компьютера на процессорах Pentium и выше. При этом возможны конкурирующие обращения к

памяти как со стороны процессора, так и со стороны мостов шин PCI.

Порт AGP может работать как в своем «естественном» режиме с конвейеризацией и сдвоенными передачами, так и в режиме шины PCI. В конвейеризированном режиме, в котором начало цикла отмечается сигналом PIPE#, возможны только обращения к памяти. В режиме PCI циклы начинаются с сигнала FRAME#, и обращения возможны как к пространству памяти, так и к пространству ввода/вывода и конфигурационному пространству. Кроме собственно AGP, в порте AGP заложены сигналы шины USB, которую предполагается заводить в монитор. Внешне карты с портом AGP похожи на PCI, но у них используется разъем повышенной плотности с «двухэтажным» (как у EISA) расположением ламелей. Сам разъем располагается дальше от задней кромки платы, чем разъем PCI.

Фирма Intel ввела поддержку AGP в чипсеты для процессоров Pentium Pro и Pentium II. Однако нет «противопоказаний» для применения AGP с Pentium.

AGP строился исключительно для графического акселератора. Порт позволяет работать в двух режимах — DMA и исполнения (*Executive Mode*). В *режиме DMA* акселератор при вычислениях рассматривает видеопамять как первичную, а когда ее недостаточно, подкачивает данные из основной памяти. При этом для трафика порта характерны длительные блочные передачи. В *режиме DIME* (*DIrect Memory Execute*) видеопамять и основная память для акселератора равнозначны и располагаются в едином адресном пространстве. Трафик порта при этом будет насыщен короткими произвольными запросами.

Отметим, что многие преимущества AGP носят потенциальный характер и могут быть реализованы лишь при поддержке аппаратных средств графического адаптера и ПО. Что касается большой пропускной способности AGP, то следует напомнить, что в графических адаптерах с памятью WRAM или RDRAM внутренняя скорость обмена данными акселератора с видеопамятью достигает 1,6 Гбайт/с, так что DIME выигрывает только за счет объема доступной памяти.

4.6. Шины PCMCIA (PC Card)

Организация PCMCIA (Personal Computer Memory Card International Association — международная ассоциация производителей карт памяти для персональных компьютеров) ввела ряд стандартов на шины расширения блокнотных компьютеров. Первый из них назывался *PCIMCIA*. Впоследствии он был переименован в *PC Card*. Шина PC Card позволяет подключать расширители памяти, модемы, контроллеры дисков и стриммеров, SCSI-адаптеры, сетевые адаптеры и т. д. Недостаточно строгое следование производителей этому стандарту приводит к проблемам совместимости. Назначение контактов разъема шины приведено в табл. 4.9. Шина адресует 64 Мбайт памяти, разрядность данных 16 бит, частота до 33 МГц, DMA и Bus-Mastering не поддерживаются. Теоретически допускается до 4080 слотов PC Card. Шина ориентирована на программное конфигурирование адаптеров. Большинство адаптеров выпускаются с поддержкой PnP и предусматривают «горячее» подключение — интерфейсные карты могут вставляться и выниматься без выключения компьютера. Для этого контакты шин питания имеют большую длину, чем сигнальные, чем обеспечивается их упреждающее подключение и запаздывающее отключение. Два контакта обнаружения карты (Card Detect) короче остальных. Все устройства имеют свою поддержку BIOS. Несмотря на возможность динамического конфигурирования, в некоторых случаях при изменении конфигурации требуется перезагрузка системы.

Различают 4 типа PC Card. Электрически идентичные, они варьируются по габаритам при совместимости снизу вверх (меньшие адаптеры встают в большие гнезда). Адаптер типа 1 имеет размеры 54×85 мм и толщину не более 3,3 мм, типа 2 — 48×75 мм и толщину 5 мм, типа 3 толщину 10,5 мм (однако HDD типа 3 имеет толщину 13 мм!).

Все устройства PC Card имеют минимальное энергопотребление. Существуют предпосылки для введения этой шины как дополнительной в настольные PC.

Шина *CardBus* с тем же 68-контактным разъемом обеспечивает расширение разрядности данных до 32 бит за счет муль-

типлексирования шины адреса и данных, обеспечивая обратную совместимость с PC Card.

Таблица 4.9. Разъем шины PC Card (PCMCIA)

Контакт	Сигнал	Контакт	Сигнал
1	GND	35	GND
2	Data 3	36	Card Detect 1#
3	Data 4	37	Data 11
4	Data 5	38	Data 12
5	Data 6	39	Data 13
6	Data 7	40	Data 14
7	Card Enable 1#	41	Data 15
8	Addr 10	42	Card Enable 2#
9	Out Enable#	43	Refresh
10	Addr 11	44	RFU (IOR#)
11	Addr 9	45	RFU (IOW#)
12	Addr 8	46	Addr 17
13	Addr 13	47	Addr 18
14	Addr 14	48	Addr 19
15	WrEnable#/Prog#	49	Addr 20
16	Ready/Busy# (IREQ)	50	Addr 21
17	+5 B	51	+5 B
18	Vpp1	52	Vpp2
19	Addr 16	53	Addr 22
20	Addr 15	54	Addr 23
21	Addr 12	55	Addr 24
22	Addr 7	56	Addr 25
23	Addr 6	57	RFU
24	Addr 5	58	Reset
25	Addr 4	59	WAIT#
26	Addr 3	60	RFU (INPACK#)
27	Addr 2	61	Register Select#
28	Addr 1	62	BatVDet2 (SPKR#)
29	Addr 0	63	BatVDet1 (STSCHG#)
30	Data 0	64	Data 8
31	Data 1	65	Data 9
32	Data 2	66	Data 10
33	WRProt/(IOCS16#)	67	Card Detect 2#
34	GND	68	GND

Для карт памяти (динамической, статической, постоянной и флэш-памяти) существует стандарт *Miniature Card*, представляющий подмножество шины PC Card. Миниатюрная карта размером 33×38×3,5 мм с 60-контактным разъемом через переходный адаптер может устанавливаться и в слот PC Card типа 2.

5. Шина SCSI

Системный интерфейс малых компьютеров *SCSI* (Small Computer System Interface, произносится «сказн») был стандартизован ANSI в 1986 году (X3.131-1986). Интерфейс предназначен для соединения устройств различных классов — памяти прямого (жесткие диски) и последовательного (стриммеры) доступа, CD-ROM, оптических дисков однократной и многократной записи, устройств автоматической смены носителей информации, принтеров, сканеров, коммуникационных устройств и процессоров. *Устройством SCSI — SCSI Device* — называется как *хост-адаптер*, связывающий шину SCSI с какой-либо внутренней шиной компьютера, так и *контроллер целевого устройства — target controller*, с помощью которого оно подключается к шине SCSI. С точки зрения шины все устройства могут быть равноправными и являться как *инициаторами обмена (инициализирующими устройствами, ИУ)*, так и *целевыми устройствами (ЦУ)*, однако чаще всего в роли ИУ выступает хост-адаптер. К одному контроллеру может подключаться несколько ПУ, по отношению к которым контроллер может быть как внутренним, так и внешним. Широкое распространение получили ПУ со встроенным контроллером SCSI (embeded SCSI controller), к которым относятся накопители на жестких магнитных дисках, CD-ROM, стриммеры. Каждое ЦУ может содержать до 8 независимо адресуемых *логических устройств* (ЛУ) со своими номерами *LUN (Logical Unit Number)*, представляющими ПУ или их части.

По физической реализации интерфейс является 8-битной параллельной шиной с тактовой частотой 5 МГц. Скорость передачи данных достигает 5 Мбайт/с. Впоследствии появилась спецификация — *SCSI-2* (X3.131-1994), расширяющая возможности шины. Тактовая частота шины *Fast* (быстрый) *SCSI-2* достигает 10 МГц, а *Ultra SCSI-2* — 20 МГц. Разрядность данных может быть увеличена до 16 бит — эта версия называется *Wide* (широкий) *SCSI-2*, а 8-битную вер-

сию стали называть *Narrow* (узкий). 16-битная шина допускает 16 устройств. Стандарт SCSI-2 определяет 32-битную версию интерфейса, но такие устройства обладают неоправданно высокой стоимостью интерфейса. Спецификация SCSI-2 определяет систему команд, которая включает набор базовых команд *CCS* (Common Command Set), обязательных для всех ПУ, и специфических команд для периферии различных классов. Стандарт полностью описывает протокол взаимодействия устройств, включая структуры передаваемой информации. Поддержка устройствами исполнения цепочек команд (до 256 команд) и независимость их работы друг от друга обусловливают высокую эффективность применения SCSI в многозадачных системах. Возможность присутствия на шине более одного контроллера (инициатора обмена) позволяет обеспечить разделяемое использование периферии несколькими компьютерами, подключенными к одной шине.

SCSI-3 — дальнейшее развитие стандарта, направленное на увеличение количества подключаемых устройств, расширение системы команд и поддержку Plug and Play. В качестве альтернативы параллельному интерфейсу SPI (SCSI-3 Parallel Interface) появляется возможность применения последовательного, в том числе волоконно-оптического, интерфейса со скоростью 100 Мбайт/с. SCSI-3 существует в виде широкого спектра документов, определяющих отдельные аспекты интерфейса. *Архитектурная модель SAM* (SCSI-3 Architecture Model) изображена на рис. 5.1. Первичный набор *общих команд SCP* (SCSI-3 Primary Commands) для устройств различных классов дополняется набором *команд* соответствующего *класса устройств*:

- *SBC* (SCSI-3 Block Commands) — для устройств памяти прямого доступа,
- *SSC* (SCSI-3 Stream Commands) — для устройств памяти последовательного доступа,
- *SGC* (SCSI-3 Graphic Commands) — для принтеров и сканеров,
- *SMC* (SCSI-3 Medium Changer Commands) — для устройств смены носителей,

циклов данных в пакете может быть неявно ограничено таймером, определяющим максимальное время, в течение которого ИУ может пользоваться шиной. ИУ завершает транзакцию одним из следующих способов:

- Нормальное завершение выполняется по окончании обмена данными.
- Завершение по тайм-ауту (*Time-out*) происходит, когда во время транзакции у ИУотбирают право на управление шиной (снятием сигнала GNT#) или когда истекает время, указанное в его таймере MLT (медленное ЦУ или слишком длинная транзакция).
- Транзакция отвергается (*Abort*), когда в течение заданного времени ИУ не получает ответа ЦУ (DEVSEL#).

Транзакция может быть прекращена и по инициативе ЦУ, для этого оно может ввести сигнал STOP#. Возможны три типа прекращения:

- Отключение (*Disconnect*) – сигнал STOP# вводится во время активности TRDY#. В этом случае транзакция завершается после фазы данных.
- Отключение с повтором (*Disconnect/Retry*) – сигнал STOP# вводится при пассивном состоянии TRDY#, и последняя фаза данных отсутствует. Является указанием ИУ на необходимость повтора транзакции.
- Отказ (*Abort*) – сигнал STOP# вводится одновременно со снятием DEVSEL# (в предыдущих случаях во время появления сигнала STOP# сигнал DEVSEL# был активен). В этом случае последняя фаза данных тоже отсутствует, но повтор не запрашивается.

Протокол квитирования обеспечивает *надежность обмена* – ИУ всегда получает информацию об отработке транзакции ЦУ. Средством повышения надежности (достоверности) является применение контроля паритета: линии AD[31:0] и C/BE#[3:0] и в фазе адреса, и в фазе данных защищены *битом паритета* PAR (количество единичных бит этих линий, включая PAR, должно быть четным). Действительное значение PAR появляется на шине с задержкой в один такт относительно линий AD и C/BE#. При обнаружении ошибки ЦУ со сдви-

гом на такт вырабатывается сигнал PERR#. В подсчете паритета при передаче данных учитываются все байты, включая и недействительные (отмеченные высоким уровнем сигнала C/BE#i). Состояние бит и недействительных байт данных во время фазы данных должно оставаться стабильным.

Арбитражем запросов на использование шины занимается специальный узел, входящий в чипсет системной платы. Каждое ИУ имеет пару сигналов — REQ# для запроса на управление шиной и GNT# для подтверждения предоставления управления шиной. Схема приоритетов (фиксированный, циклический, комбинированный) определяется программированием арбитра.

Каждое устройство — потенциальное ИУ (*PCI Master*) — имеет собственный программируемый *таймер MLT* (*Master Latency Timer*), определяющий максимальное количество тактов шины, допустимое для одной транзакции. Его конфигурированием осуществляется распределение полосы пропускания шины между устройствами.

Каждое ЦУ имеет *инкрементный механизм слежения* за длительностью циклов (*Incremental Latency Mechanism*), который не позволяет интервалу между соседними фазами данных в пакете превышать 8 тактов шины. Если ЦУ не успевает работать в таком темпе, оно обязано остановить транзакцию.

Адресация памяти, портов и конфигурационных регистров различна. Байты шины AD, несущие действительную информацию, выбираются сигналами C/BE[3:0] в фазах данных (внутри пакета эти сигналы могут менять состояние). В циклах обращения к памяти адрес, выровненный по границе двойного слова, передается по линиям AD[31:2], линии AD[1:0] задают порядок чередования адресов в пакете:

- 00 — линейное инкрементирование;
- 01 — чередование адресов с учетом длины строки кэш-памяти;
- 1x — зарезервировано.

В циклах обращения к портам ввода/вывода для адресации любого байта используются все линии AD[31:0]. В циклах конфигурационной записи/считывания устройство выбира-

- *SCC* (SCSI-3 Controller Commands) – для хост-контроллеров.

Транспортный уровень может использовать различные протоколы с соответствующей поддержкой *физических соединений*:

- *SIP* (SCSI-3 Interlocked Protocol) – протокол обмена традиционного интерфейса, физически реализуемый интерфейсом *SPI*.
- *FCP* (Fibre Channel Protocol) – протокол оптоволоконного канала с соответствующим физическим уровнем *FC-PH*.
- *SBP* (Serial Bus Protocol) – протокол последовательной шины, реализуемый интерфейсом 1394 (FireWire).
- *GPP* (Generic Packetized Protocol) – обобщенный пакетный протокол, реализуемый любым пакетным интерфейсом.
- *SSP* (Serial Storage Protocol) – последовательный протокол памяти, реализованный на архитектуре последовательной памяти SSA (Serial Storage Architecture).

К примеру, дисковый накопитель SCSI-3 с параллельным интерфейсом описывает набор стандартов SPI+SIP+SAM+SPC+SBC, а для того же устройства, но с последовательным интерфейсом вместо SPC+SBC будет связка FCP+FC-PH.

Заявка о поддержке устройством стандарта SCSI-3 непосредственно на повышение производительности по сравнению со SCSI-2 не указывает. Однако устройства SCSI-3 в большинстве случаев показывают более высокую производительность.

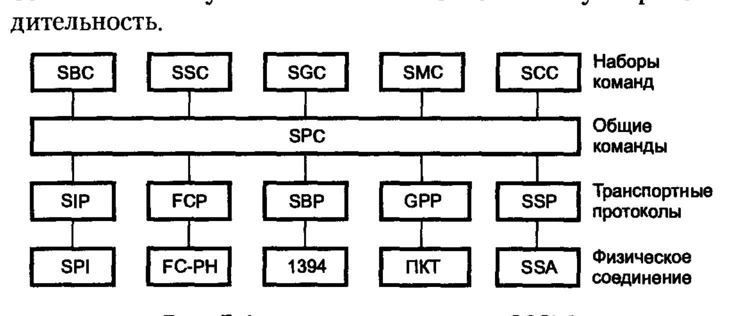


Рис. 5.1. Архитектурная модель SCSI-3

Для параллельных шин скорость передачи данных определяется *частотой передач*, измеряемой в миллионах передач в секунду — МТ/с (Mega Transfer/sec), и *разрядностью*. Комбинации частоты и разрядности обеспечивают широкий диапазон пропускной способности (табл. 5.1), достигающей 80 Мбайт/с для версии *Ultra2 Wide SCSI*. *SCSI Fast* означает частоту передач 10 МТ/с, временные диаграммы для такого режима определены в SCSI-2. Название *SCSI Fast-20* указывает на частоту передач 20 МТ/с. Этот режим, более известный как *SCSI Ultra*, определен для параллельного интерфейса в SCSI-3. *SCSI Fast-40* указывает на частоту передач 40 МТ/с. Этот режим, определенный в SCSI-3 и называемый *Ultra2 SCSI*, в настоящее время является самым быстрым для параллельной шины. Он реализован только в низковольтной дифференциальной версии интерфейса — LVD. В SCSI-3 предусмотрен режим *SCSI Fast-80*.

Для Narrow SCSI использовался разъем, изображенный на рис. 5.2. *Wide SCSI* использует разъем, изображенный на рис. 5.3. Для устройств с «горячей» заменой применяют миниатюрный D-образный разъем SCA-2, общий для питания и сигнальных цепей (рис. 5.4).

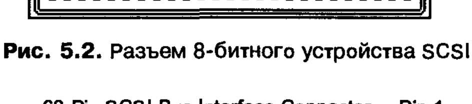


Рис. 5.2. Разъем 8-битного устройства SCSI

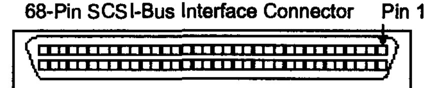


Рис. 5.3. Разъем 16-битного устройства SCSI

80-Pin SCSI SCA Connector

Pin 1

Рис. 5.4. Разъем устройства SCSI с «горячей» заменой

Таблица 5.1. Скорость передачи данных по параллельной шине SCSI(Мбайт/с)

Разрядность шины, бит	Разновидность			
	Обычный	Fast	Fast-20 (Ultra)	Fast-40 (Ultra2)
8 (Narrow)	5	10	20	40
16 (Wide)	10	20	40	80
32 (Wide)*	20	40	80	160

* Реализации не встречаются.

Последовательный интерфейс *FCAL* (Fibre Channel Arbitrated Loop — кольцо волоконного канала с арбитражем) по реализации ближе к интерфейсам локальных сетей. Этот интерфейс, известный также как *Fibre Channel SCSI*, может иметь как электрическую (коаксиальный кабель), так и оптоволоконную реализацию. В обоих случаях частота 1 ГГц обеспечивает скорость передачи данных 100 Мбайт/с. Медный кабель допускает длину шины до 30 м, оптический — до 10 км. Здесь используются иные протокольный и физический уровни интерфейса и имеется возможность подключения 126 устройств.

В настоящее время наибольшее распространение имеют устройства SCSI-2, которые сохраняют совместимость с исходной версией, теперь называемой *SCSI-1*. Однако смешивать устройства SCSI-1 и SCSI-2 неэффективно, да и не всегда возможно из-за проблем, о которых речь пойдет далее. Для краткости номер версии SCSI будем опускать, по умолчанию подразумевая Narrow SCSI-2. На ее примере разберем работу интерфейса, а особенности версии *Wide* отметим отдельно.

5.1. Физический интерфейс

Физически SCSI представляет собой шину, состоящую из 25 сигнальных цепей. Для защиты от помех каждая сигнальная цепь имеет свой отдельный обратный провод. На применяемых двухрядных разъемах контакты сигнальных и обратных цепей располагаются друг против друга. Это позволяет применять в качестве кабелей как витые пары проводов, так и плоские шлейфы, где сигнальные и обратные провода чередуются.

По типу сигналов различают *линейные* (Single Ended) и *дифференциальные* (Differential) версии SCSI. Их кабели и разъемы идентичны, но ***электрической совместимости устройств нет***. Символические обозначения для разных версий приведены на рис. 5.5.

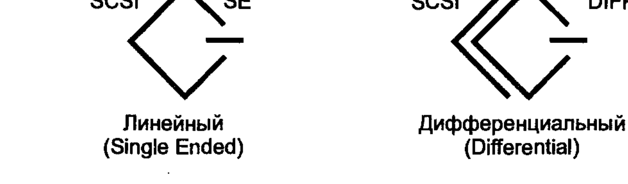


Рис. 5.5. Универсальные символические обозначения версий SCSI

Дифференциальная версия для каждой цепи задействует пару проводников, по которым передается парафазный сигнал. Здесь используются специальные дифференциальные приемопередатчики, применяемые и в интерфейсе RS-485, что позволяет значительно увеличить длину кабеля, сохраняя частоту обмена (табл. 5.2). Дифференциальный интерфейс применяется в дисковых системах серверов. Традиционный дифференциальный интерфейс получил название «высоковольтный» — High Voltage Differential (HVD), поскольку в SCSI-3 ему появилась низковольтная альтернатива — Low Voltage Differential (LVD). Низковольтный вариант достигает частоты 40 МТ/с в устройствах Ultra2 SCSI при длине шины 25 м (8 устройств) или 12 м (16 устройств).

В широко используемой *линейной* версии каждый сигнал передается потенциалом с ТТЛ-уровнями относительно общего провода. Здесь общий (обратный) провод для каждого сигнала тоже должен быть отдельным. В дальнейшем речь пойдет об этой версии.

Новые устройства с интерфейсом LVD могут работать на шине вместе с устройствами с линейным интерфейсом — для этого их буферные схемы содержат автоматический определитель типа интерфейса. Однако совместимость относится только к LVD — традиционные устройства с HVD могут работать только с себе подобными.

Плоский кабель используется для соединения устройств, расположенных в одном корпусе. На нем может быть наколото несколько разъемов. При необходимости кабели могут сращиваться через специальные переходные разъемы. Кабели сращиваются только через концевые разъемы, Т-образные ответвления недопустимы. *Круглый кабель*, состоящий из витых пар, используется для соединения вне корпусов устройств. ПУ внешнего исполнения обычно имеют два разъема, что позволяет соединить их в цепочку. Длина кабеля зависит от версии интерфейса и частоты (см. табл. 5.2). При подсчете суммарной длины кабеля следует учитывать возможность использования одного порта хост-адаптера одновременно для внешних и внутренних подключений и суммировать длины внутренних и внешних кабелей.

Таблица 5.2. Максимальная длина кабелей SCSI

Тип интерфейса	Обычный (5 МТ/с)	Fast (10 МТ/с)	Ultra (20 МТ/с)	Ultra 2 (40 МТ/с)
Линейный (Single ended)	6 м	3 м	1,5 м	–
Дифференциальный (HVD)	25 м	12 м	6 м	–
Дифференциальный низковольтный (LVD)	25 м	25 м	25 м	25 м (8 устройств) 12 м (16 устройств)

Ассортимент кабелей SCSI довольно широк:

- *А-кабель*: стандартный для 8-битного интерфейса, 25 пар проводов. Для внутренних устройств используется плоский шлейф с разъемами IDC-50, для внешних — экранированный круглый кабель с разъемами *CENTRONICS-50*.
- *В-кабель*: малораспространенный 16/32-битный расширитель SCSI-2.
- *Р-кабель*: 8/16-битный кабель с 34 парами проводов, снабжен улучшенными миниатюрными экранированными разъемами. Применяется в интерфейсах SCSI-2/3, в 8-битном варианте контакты 1–5, 31–39, 65–68 не используются. Разъемы для внешнего подключения выглядят как

миниатюрный вариант *Centronics* с плоскими контактами, внутренние имеют штырьковые контакты.

- *Q-кабель*: 68-проводное расширение до 32 бит, используется в паре с P-кабелем и имеет аналогичную конструкцию.
- *Кабель с разъемами DB-25P* — 8-битный, стандартный для Macintosh (см. табл. 5.5), используется на некоторых внешних устройствах (Iomega ZIP Drive). Встречается иная раскладка цепей, если 25-контактный разъем установлен на хост-адаптере.

Возможны также различные варианты кабелей-переходников.

Назначение контактов разъемов кабелей приведено в табл. 5.3–5.6. Неудобство вызывает система нумерации контактов, которая различна для внешних и внутренних разъемов. Однако физическая раскладка проводов на разъеме и в плоском кабеле одинакова.

Таблица 5.3. Разъемы A-кабеля SCSI

Контакт разъема		Сигнал	Контакт разъема		Сигнал
Внутреннего	Внешнего		Внутреннего	Внешнего	
1	1	GND	2	26	DB0#
3	2	GND	4	27	DB1#
5	3	GND	6	28	DB2#
7	4	GND	8	29	DB3#
9	5	GND	10	30	DB4#
11	6	GND	12	31	DB5#
13	7	GND	14	32	DB6#
15	8	GND	16	33	DB7#
17	9	GND	18	34	DBP0#
19	10	GND	20	35	GND
21	11	GND	22	36	GND
23	12	Reserved	24	37	Reserved
25	13	Open	26	38	TERMPWR
27	14	Reserved	28	39	Reserved

Таблица 5.3 (*продолжение*)

Контакт разъема		Сигнал	Контакт разъема		Сигнал
Внутреннего	Внешнего		Внутреннего	Внешнего	
29	15	GND	30	40	GND
31	16	GND	32	41	ATN#
33	17	GND	34	42	GND
35	18	GND	36	43	BSY#
37	19	GND	38	44	ACK#
39	20	GND	40	45	RST#
41	21	GND	42	46	MSG#
43	22	GND	44	47	SEL#
45	23	GND	46	48	C/D#
47	24	GND	48	49	REQ#
49	25	GND	50	50	I/O#

Таблица 5.4. Разъемы В-кабеля SCSI

Контакт разъема		Сигнал	Контакт разъема		Сигнал
Внутреннего	Внешнего		Внутреннего	Внешнего	
1	1	GND	2	35	GND
3	2	GND	4	36	DB8#
5	3	GND	6	37	DB9#
7	4	GND	8	38	DB10#
9	5	GND	10	39	DB11#
11	6	GND	12	40	DB12#
13	7	GND	14	41	DB13#
15	8	GND	16	42	DB14#
17	9	GND	18	43	DB15#
19	10	GND	20	44	DBP1#
21	11	GND	22	45	ACKB#
23	12	GND	24	46	GND
25	13	GND	26	47	REQB#
27	14	GND	28	48	DB16#

Контакт разъема		Сигнал	Контакт разъема		Сигнал
Внутреннего	Внешнего		Внутреннего	Внешнего	
29	15	GND	30	49	DB17#
31	16	GND	32	50	DB18#
33	17	TERMPWR	34	51	TERMPWR
35	18	TERMPWR	36	52	TERMPWR
37	19	GND	38	53	DB19#
39	20	GND	40	54	DB20#
41	21	GND	42	55	DB21#
43	22	GND	44	56	DB22#
45	23	GND	46	57	DB23#
47	24	GND	48	58	DBP2#
49	25	GND	50	59	DB24#
51	26	GND	52	60	DB25#
53	27	GND	54	61	DB26#
55	28	GND	56	62	DB27#
57	29	GND	58	63	DB28#
59	30	GND	60	64	DB29#
61	31	GND	62	65	DB30#
53	32	GND	64	66	DB31#
65	33	GND	66	67	DBP2#
67	34	GND	68	68	GND

Таблица 5.5. Разъем SCSI DB-25

Контакт	Сигнал	Контакт	Сигнал	Контакт	Сигнал
1	REQ#	10	DB3#	19	SEL#
2	MSG#	11	DB5#	20	DBP0#
3	I/O#	12	DB6#	21	DB1#
4	RST#	13	DB7#	22	DB2#
5	ACK#	14	GND	23	DB4#
6	BSY#	15	C/D#	24	GND
7	GND	16	GND	25	TERMPWR
8	DB0#	17	ATN#		
9	GND	18	GND		

Таблица 5.6. Разъемы P-, Q-кабелей SCSI

Контакт	P- и Q-кабели, сигнал	Контакт	P-кабель, сигнал	Q-кабель, сигнал
1	GND	35	DB12#	DB28#
2	GND	36	DB13#	DB29#
3	GND	37	DB14#	DB30#
4	GND	38	DB15#	DB31#
5	GND	39	DBP1#	DBP3#
6	GND	40	DB0#	DB16#
7	GND	41	DB1#	DB17#
8	GND	42	DB2#	DB18#
9	GND	43	DB3#	DB19#
10	GND	44	DB4#	DB20#
11	GND	45	DB5#	DB21#
12	GND	46	DB6#	DB22#
13	GND	47	DB7#	DB23#
14	GND	48	DBP0#	DBP2#
15	GND	49	GND	GND
16	GND	50	GND	GND
17	TERMPWR	51	TERMPWR	TERMPWRQ
18	TERMPWR	52	TERMPWR	TERMPWRQ
19	Reserved	53	Reserved	Reserved
20	GND	54	GND	GND
21	GND	55	ATN#	Terminated
22	GND	56	GND	GND
23	GND	57	BSY#	Terminated
24	GND	58	ACK#	ACKQ#
25	GND	59	RST#	Terminated
26	GND	60	MSG#	Terminated
27	GND	61	SEL#	Terminated
28	GND	62	C#/D	Terminated
29	GND	63	REQ#	REQQ#

Контакт	P- и Q-кабели, сигнал	Контакт	P-кабель, сигнал	Q-кабель, сигнал
30	GND	64	I/O#	Terminated
31	GND	65	DB8#	DB24#
32	GND	66	DB9#	DB25#
33	GND	67	DB10#	DB26#
34	GND	68	DB11#	DB27#

Назначение сигналов раскрывает табл. 5.7. Все сигналы шины являются *L-активными*: активному состоянию и логической единице соответствует низкий потенциал. На концах кабельных шлейфов обязательно устанавливаются *терминаторы*, согласованные по импедансу с кабелем. Они предназначены для «подтягивания» уровня сигналов линий к высокому потенциалу. Терминаторы служат и для предотвращения отражения сигналов от концов кабеля. По исполнению терминаторы могут быть *внутренние* (размещенные на печатной плате устройства) и *внешние* (устанавливаемые на разъемы кабеля или устройства). По электрическим свойствам различают следующие типы терминаторов:

- *Пассивные* (SCSI-1) с импедансом 132 Ом, представляющие собой обычные резисторы. Не пригодны для режимов SCSI-2 с частотой выше 5 МГц.
- *Активные* с импедансом 110 Ом — специальные терминаторы для работы на частоте 10 МГц.
- *FPT* (Forced Perfect Terminator) — улучшенный вариант активных терминаторов с ограничителями выбросов.

Активные терминаторы требуют питания, для чего в интерфейсе имеются специальные линии TERMPWR. Питание поступает от ИУ.

Таблица 5.7. Назначение сигналов шины SCSI

Сигнал	Назначение
BSY#	*Busy* — шина занята
SEL#	*Select* — выбор ЦУ инициатором (*Select*) или инициатора ЦУ (*Reselect*)

Сигнал	Назначение
C/D#	*Control/Data* – управление (низкий уровень)/данные (высокий уровень)
I/O#	*Input/Output* – направление передачи относительно ИУ: вводу в ИУ соответствует низкий уровень. Используется для различия прямой (*Select*) и обратной (*Reselect*) выборки: фазе *Selection* соответствует низкий уровень
MSG#	*Message* – передача сообщения
DB[0:31]#	*Data Bus* – инверсная шина данных
DP[0:3]#	*Data Parity* – инверсные биты паритета, дополняют количество единичных бит байта до нечетного. DP0# относится к DB[0:7], ... DP3# – к DB[24:31]. В фазе арбитража не действуют
TERMPWR	*Terminator Power* – питание терминаторов
ATN#	*Attention* – внимание
REQ#	*Request* – запрос от ЦУ на пересылку данных
ACK#	*Acknowledge* – подтверждение передачи (ответ на REQ#)
RST#	*Reset* – сброс

Каждое устройство SCSI, подключенное к шине, должно иметь свой уникальный *адрес*, назначаемый при конфигурировании. Для 8-битной шины диапазон значений адреса 0–7, для 16-битной – 0–15. Адрес задается предварительной установкой переключателей или джамперов. Для хост-адаптера возможно программное конфигурирование. Адресация устройств на шине в фазах выборки осуществляется через *идентификатор SCSI ID*, представляющий адрес в позиционном коде. Адрес определяет номер линии шины данных, которой осуществляется выборка данного устройства. Устройство с нулевым адресом выбирается низким уровнем на линии DB0# (SCSI ID=00000001), с адресом 7 – на линии DB7# (SCSI ID=10000000). Для ИУ значение идентификатора определяет приоритет устройства при использовании шины, наибольший приоритет имеет устройство с большим значением адреса. Адрес и идентификатор – всего лишь две различные формы представления одного и того же параметра.

В любой момент обмен информацией по шине может происходить только между парой устройств. Операцию начинает ИУ, а ЦУ ее исполняет. ИУ выбирает ЦУ по его идентификатору. Чаще всего роли устройств фиксированы: хост-адаптер является инициатором (ИУ), а ПУ – целевым (ЦУ). Возможны комбинированные устройства, выступающие в роли и ИУ, и ЦУ. В ряде случаев роли устройств меняются: ЦУ может, пройдя фазу арбитража, выполнить обратную выборку (*Reselect*) ИУ для продолжения прерванной операции. При выполнении команды копирования (*Copy*) ИУ дает указание ***ведущему устройству копирования*** (*Copy Master*) на обмен данными, который может производиться и с другим ЦУ (для которых ведущее устройство копирования выступит в роли ИУ).

Информация по шине данных передается побайтно *асинхронно*, используя механизмы запросов (REQuest) и подтверждений (ACKnowledge). Каждый байт контролируется на нечетность (кроме фазы арбитража), но контроль может быть отключен. Интерфейс имеет возможность *синхронной* передачи данных, ускоряющей обмен.

5.2. Фазы шины

Шина может находиться в одной из перечисленных *фаз*. Роли источников сигналов между ИУ и ЦУ описаны в табл. 5.8.

Таблица 5.8. Источники сигналов SCSI

Фаза шины	Сигнал				
	BSY#	SEL#	REQ#, C/D#, I/O#, MSG#	ACK#, ATN#	DBx#, DBPx#
Bus Free	–	–	–	–	–
Arbitration	AA	WA	–	–	SID
Selection	I, T	I	–	I	I
Reselection	I, T	T	T	I	T
Command	T	–	T	I	I

Фаза шины	Сигнал				
	BSY#	SEL#	REQ#, C/D#, I/O#, MSG#	ACK#, ATN#	DBx#, DBPx#
Data IN	T	–	T	I	T
Data OUT	T	–	T	I	I
Status	T	–	T	I	T
Message IN	T	–	T	I	T
Message OUT	T	–	T	I	I

I – источник сигнала – ИУ;

T – источник сигнала – ЦУ;

AA – источник сигнала – устройство, активное в арбитраже;

WA – источник сигнала – устройство-победитель в арбитраже;

SID – каждое устройство управляет только битом данных, соответствующим значению его SCSI ID.

В фазе *Bus Free* шина находится в состоянии покоя – нет никаких процессов обмена; она готова к арбитражу. Признаком является пассивное состояние линий BSY# и SEL#.

В фазе *Arbitration* устройство может получить право на управление шиной. Дождавшись покоя шины (*Bus Free*), устройство вводит сигнал BSY# и свой идентификатор SCSI ID. Если идентификаторы выставили несколько устройств одновременно, то право на управление шиной получает устройство с наибольшим адресом, а остальные устройства отключаются до следующего освобождения шины. Устройство, выигравшее арбитраж, вводит сигнал SEL# и переходит в фазу *Selection* или *Reselection*.

В фазе *Selection* ИУ, выигравшее арбитраж, вводит на шину данных результат логической функции ИЛИ от пары идентификаторов – своего и ЦУ, – сопровождая его битом паритета. Установкой сигнала ATN# ИУ указывает, что следующей фазой будет *Message OUT*. ИУ снимает сигнал BSY#. Отсутствие сигнала I/O# отличает данную фазу от *Reselection*. Адресованное ЦУ отвечает сигналом BSY#, если паритет кор-

ректный и на шине данных присутствует только пара идентификаторов (его и ИУ). На некорректные значения данных устройства отвечать не должны. Если за заданное время ЦУ не ответило, срабатывает тайм-аут, ИУ освобождает шину или вводит сигнал сброса RST#.

Фаза *Reselection* аналогична предыдущей, но ее вводит ЦУ. Фаза появляется в том случае, когда ЦУ на время исполнения команды отключалось от шины. По завершении внутренней операции это устройство, выиграв арбитраж, будет вызывать ИУ, которое ранее породило исполнение операции. ЦУ снимает сигнал BSY#, активность сигнала I/O# отличает данную фазу от фазы *Selection*. Адресованное ИУ отвечает сигналом BSY#, условия ответа и тайм-аут аналогичны предыдущей фазе.

В фазах *Command, Data, Status* и *Message* по шине данных передается информация, фазы идентифицируются сигналами MSG#, C/D# и I/O# (табл. 5.9), которыми управляет ЦУ. ИУ может потребовать посылки сообщения (фаза *Message OUT*) введением сигнала ATN#, а ЦУ может освободить шину, сняв сигналы MSG#, C/D#, I/O# и BSY#.

Таблица 5.9. Информационные фазы SCSI

Сигнал			Фаза	Направление
MSG#	C/D#	I/O#		
0	0	0	*Data OUT*	I ⇨ T
0	0	1	*Data IN*	I ⇦ T
0	1	0	*Command*	I ⇨ T
0	1	1	*Status*	I ⇦ T
1	0	0	Зарезервировано	
1	0	1	Зарезервировано	
1	1	0	*Message OUT*	I ⇨ T
1	1	1	*Message IN*	I ⇦ T

Временные диаграммы *асинхронного обмена* приведены на рис. 5.6. Здесь передача каждого байта сопровождается взаимосвязанной парой сигналов REQ#/ACK#. ИУ фиксирует принимаемые данные, получив сигнал REQ# (по отрицатель-

ному перепаду). ЦУ считает принимаемые данные действительными по отрицательному перепаду сигнала ACK#. Асинхронный обмен поддерживается всеми устройствами для всех фаз передачи информации.

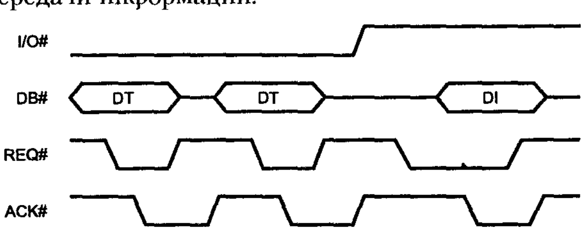


Рис. 5.6. Временные диаграммы асинхронного обмена (DI — данные от ИУ, DT — данные от ЦУ)

Фазы передачи данных *Data OUT* и *Data IN* по предварительной «договоренности» устройств могут выполняться и в *синхронном режиме* обмена, диаграммы которого приведены на рис. 5.7. При согласовании синхронного режима определяются минимальные длительности и периоды управляющих импульсов ACK# и REQ#, а также допустимое отставание подтверждений от запросов (*REQ/ACK offset agreement*). ЦУ передает серию данных, сопровождаемых стробами REQ# (рис. 5.7а) в темпе, ограниченном установленными временными параметрами. ИУ фиксирует принимаемые данные по отрицательному перепаду сигнала REQ#, но отвечать на них сигналом ACK# может с некоторым опозданием. Как только отставание числа принятых сигналов ACK# от числа посланных REQ# достигнет оговоренного предельного значения (в данном примере 2), ЦУ приостановит обмен до прихода очередного подтверждения ACK#. Операция будет считаться завершенной, когда число принятых подтверждений совпадет с числом посланных запросов. При приеме данных ЦУ механизм согласования остается тем же, но данные фиксируются по отрицательному перепаду сигнала ACK# (рис. 5.7б).

В спецификации SCSI-1 момент возобновления передачи по устранении отставания описан нечетко, в результате чего разработчики могли считать, что очередной запрос (и данные) может последовать лишь после окончания (положительного перепада) сигнала ACK#. Устройство, на это рассчитанное, может терять данные: для него последний сигнал REQ# (и данные) будет неожиданным и выглядеть как превышение согласованного смещения.

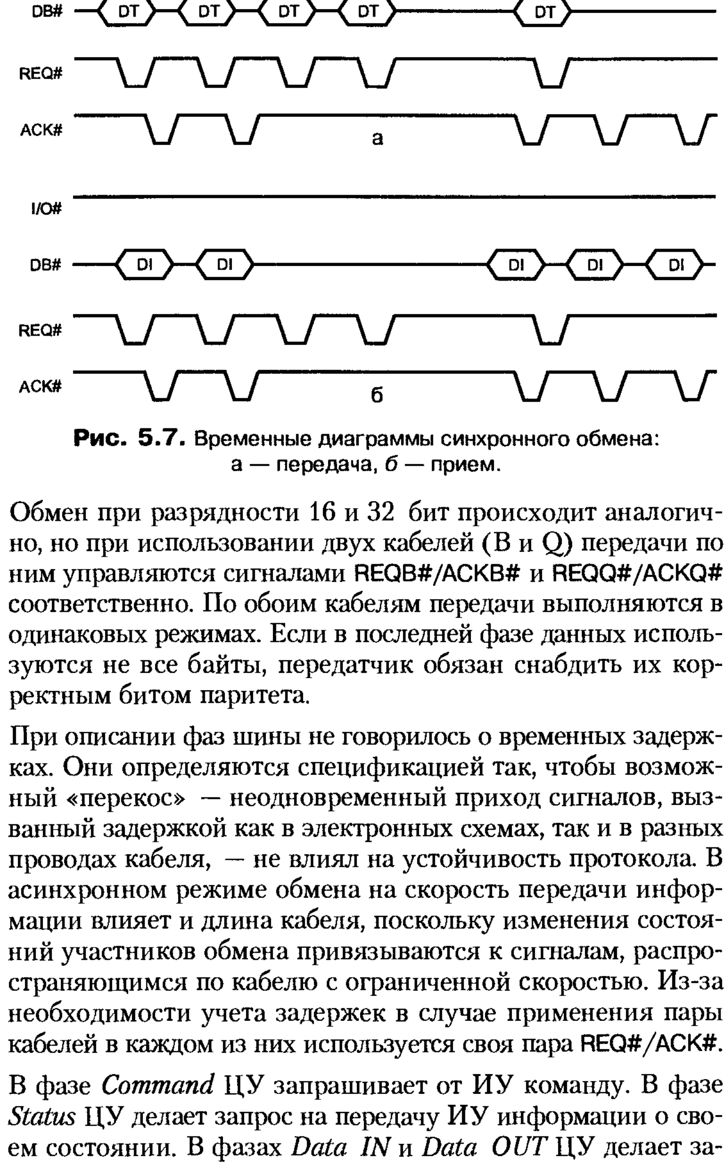


Рис. 5.7. Временные диаграммы синхронного обмена: а — передача, *б* — прием.

Обмен при разрядности 16 и 32 бит происходит аналогично, но при использовании двух кабелей (B и Q) передачи по ним управляются сигналами REQB#/ACKB# и REQQ#/ACKQ# соответственно. По обоим кабелям передачи выполняются в одинаковых режимах. Если в последней фазе данных используются не все байты, передатчик обязан снабдить их корректным битом паритета.

При описании фаз шины не говорилось о временных задержках. Они определяются спецификацией так, чтобы возможный «перекос» — неодновременный приход сигналов, вызванный задержкой как в электронных схемах, так и в разных проводах кабеля, — не влиял на устойчивость протокола. В асинхронном режиме обмена на скорость передачи информации влияет и длина кабеля, поскольку изменения состояний участников обмена привязываются к сигналам, распространяющимся по кабелю с ограниченной скоростью. Из-за необходимости учета задержек в случае применения пары кабелей в каждом из них используется своя пара REQ#/ACK#.

В фазе *Command* ЦУ запрашивает от ИУ команду. В фазе *Status* ЦУ делает запрос на передачу ИУ информации о своем состоянии. В фазах *Data IN* и *Data OUT* ЦУ делает за-

просы на передачу данных к ИУ и от него соответственно. Фазы *Message IN* и *Message OUT* служат для передачи *сообщений*. Фазу *Message OUT* ЦУ вводит в ответ на *условие Attention*, порождаемое ИУ сигналом ATN#, когда оно нуждается в посылке сообщения ЦУ. Фазу *Message IN* ЦУ вводит при необходимости посылки сообщения ИУ.

Между фазами передачи информации сигналы BSY#, SEL#, REQ# и ACK# должны оставаться в неизменном состоянии, меняться могут только значения сигналов C/D#, I/O#, MSG# и шины данных.

Сигналы ATN# и RST# могут порождать *условия Attention* и *Reset* соответственно, причем асинхронно по отношению к фазам шины. Эти условия могут привести к изменению предопределенного порядка фаз. Сигнал ATN# вводится ИУ во время любой фазы, кроме арбитража и состояния покоя шины. Сигнал RST# вводится в любой момент любым устройством, и по условию *Reset* все устройства должны немедленно освободить шину. В зависимости от настройки, принятой для всех устройств конкретной системы, возможно выполнение одного из двух вариантов сброса. «Жесткий» сброс переводит устройства в состояние, принятое по включению питания, сбрасывая все текущие процессы, очереди и т. п. В случае «мягкого» сброса после освобождения шины устройства пытаются завершить начатые операции, сохраняя текущие назначения настроек.

Каждый *процесс ввода/вывода* состоит из следующей последовательности фаз шины: из состояния *Bus Free* через фазу *Arbitration* переход к фазе *Selection* или *Reselection*. Далее следуют фазы передачи информации (*Command, Data, Status, Message*). Завершающей фазой является *Message In*, в которой передается сообщение *Disconnect* или *Command Complete*, после чего шина переходит в состояние покоя *Bus Free*.

Архитектура SCSI обеспечивает для каждого процесса ввода/вывода сохранение набора из трех *указателей* — *Saved SCSI Pointers*, для команды, состояния и данных. ИУ имеет текущий набор указателей (только один), в который копируется сохраненный набор для текущего процесса. Текущие указатели указывают на очередной байт команды, состоя-

ния и данных, которые будут передаваться между памятью ИУ и ЦУ. Сохраненные указатели команды и состояния всегда указывают на начала блоков дескрипторов команд и состояния. Сохраненный указатель данных указывает на начало блока данных до тех пор, пока ЦУ не пришлет сообщение *Save Data Pointer.* По его приему будет сохранен текущий указатель данных. Когда ЦУ отключается от шины, информация о текущем процессе ввода/вывода содержится в сохраненном наборе указателей. При возобновлении процесса ЦУ сообщением *Restore Pointers* может потребовать у ИУ скопировать сохраненный набор в текущий и продолжить выполнение команд данного процесса ввода/вывода.

Поскольку указатвль данных может быть модифицирован ЦУ до завершения ввода/вывода, использование указатвля для определения реального количества переданных данных дает ненадежные результаты.

5.3. Управление интерфейсом

Для управления интерфейсом служит система сообщений — *Message System,* которыми обмениваются ИУ и ЦУ. Обмен происходит в фазах *Message IN/OUT* (см. выше), в одной фазе может передаваться несколько сообщений. Одно сообщение не может расщепляться на несколько фаз. Форматы сообщений стандартизованы; каждое сообщение начинается с кода. Существуют однобайтные (коды 00h, 02h–1Fh, 80h–FFh), двухбайтные (коды 20h–2Fh) и расширенные сообщения (код 01h). В двухбайтном сообщении второй байт является аргументом сообщения. В расширенных сообщениях второй байт задает длину, а последующие байты несут код и аргументы сообщения. Коды сообщений приведены в табл. 5.10.

Таблица 5.10. Сообщения шины SCSI

Код	Направление	Сообщение	Назначение
00h	*In*	*Command Complete*	Процесс ввода/вывода завершен, информация о состоянии послана в сообщении Status
02h	*In*	*Save Data Pointer*	Сохранение указателя данных

Таблица 5.10 (продолжение)

Код	Направление	Сообщение	Назначение
03h	*In*	*Restore Pointers*	Восстановление указателей
04h	*In*	*Disconnect*	Текущее соединение разрывается, но для продолжения процесса потребуется повторное соединенис (не вызываст сохранения текущих указателей)
04h	*Out*	*Disconnect*	Инструкция ЦУ на разрыв соединения
05h	*Out*	*Initiator Detected Error*	ИУ обнаружило ошибку на шине
06h	*Out*	*Abort*	Сброс всех процессов, связанных с данной парой I_T, и освобождение шины
07h	*In/Out*	*Message Reject*	Сообщение (или его параметры) недействительны для получателя
08h	*Out*	*No Operation*	ИУ нечего ответить на запрос сообщения
09h	*Out*	*Message Parity Error*	Последний байт сообщения принят с неверным паритетом
0Ah	*In*	*Linked Command Complete*	Цепочка команд завершена. ИУ может инициализировать указатели для новой цепочки
0Bh	*In*	*Linked Command Complete (With Flag)*	То же, но ИУ между цепочками команд вызывает прерывание в хост-системе
0Ch	*Out*	*Bus Device Reset*	Аппаратный сброс ЦУ и освобождение шины
0Dh	*Out*	*Abort Tag*	Сброс текущего процесса, связанного с данной парой I_T, и освобождение шины

Код	Направление	Сообщение	Назначение
0Eh	*Out*	*Clear Queue*	Сброс всех процессов и освобождение шины (эквивалентно серии сообщений Abort, принятых от всех ИУ)
0Fh	*In/Out*	*Initiate Recovery*	Сообщение для уведомления об асинхронных событиях (ЦУ временно становится ИУ)
10h	*Out*	*Release Recovery*	Завершение обработки асинхронных событий
11h	*Out*	*Terminate I/O Process*	Принудительное завершение текущего процесса без логического повреждения носителя у ЦУ
12h – 1Fh, 30h – 7Fh	Зарезервированы для 1-байтных сообщений		
80h -FFh	*Out*	*Identify*	Установление связи типа I_T_L или I_T_R
80h -FFh	*In*	*Identify*	Восстановление связи типа I_T_L или I_T_R при повторном соединении. Вызывает восстановление сохраненных указателей
2-байтные сообщения			
20h	*In*	*Simple Queue Tag*	Процесс помещен в очередь, аргумент – тег (00h–FFh)
20h	*Out*	*Simple Queue Tag*	Обращение к конкретному процессу маркированной очереди, аргумент – тег (00h–FFh)
21h	*Out*	*Head Of Queue Tag*	Помещение процесса в начало маркированной очереди для данного LUN, аргумент – тег (00h–FFh)
22h	*Out*	*Ordered Queue Tag*	Помещение процесса в конец маркированной очереди для данного LUN, аргумент – тег (00h–FFh)

Таблица 5.10 (*продолжение*)

Код	Направление	Сообщение	Назначение
23h	*In*	*Ignore Wide Residue*	Последние байты последнего переданного слова недействительны (аргумент см. в табл. 5.11)
24h–2Fh	Зарезервированы для 2-байтных сообщений		
Расширенные сообщения. Первый байт – 01h, в поле «код» первым байтом указана длина, вторым – расширенный код			
05h, 00h	*In*	*Modify Data Pointer*	Запрос модификации указателя данных — суммирование текущего указателя с 4-байтным аргументом сообщения
03h, 01h	*In/Out*	*Synchronous Data Transfer Request*	Параметры синхронного режима обмена. 1-й байт аргумента определяет период посылки, 2-й – допустимое смещение REQ/ACK
02h, 03h	*In/Out*	*Wide Data Transfer Request*	Разрядность передач в фазах данных. Аргумент задает число байт: 0 — 1 байт, 1 – 2 байта, 2 – 4 байта

Таблица 5.11. Игнорирование лишних данных

Значение 2-го байта	Игнорировать биты данных при разрядности обмена	
	32 бит	16 бит
01h	DB[31:24]	DB[15:8]
02h	DB[31:16]	Зарезервировано
03h	DB[31:8]	Зарезервировано
00h, 04h — FFh	Зарезервировано	Зарезервировано

В SCSI-2 для установления связи процесса с конкретным *логическим устройством I_T_L* (*Initiator_Target_LUN*) или с конкретной *целевой программой I_T_R* (*Initiator_Target_TRN*),

а также предоставления права разрыва соединения служат сообщения *Identify*. В байте их кодов биты [2:0] в зависимости от бита 5 *LUNTAR* задают номер *LUN* (*LUNTAR*=0) или *TRN* (*LUNTAR*=1). Каждый процесс может быть адресован только одному *LUN* или *TRN*. Если ЦУ во время выполнения процесса обнаружит сообщение с иным адресом связи, оно обязано освободить шину (ситуация неожиданного разрыва). Единичным значением бита 6 *DiscPriv* при передаче сообщения ИУ наделяет ЦУ правом разрыва соединения.

ИУ может проинструктировать ЦУ на *разрыв соединения*, послав ему сообщение *Disconnect*. Получив сообщение, ЦУ посылает одноименное сообщение (предварительно ЦУ может потребовать сохранения указателя данных, послав сообщение *Save Data Pointer*) и освобождает шину. Если ЦУ не поддерживает эту возможность, оно отвечает сообщением *Message Reject*. Когда процесс, продолжающийся в устройстве, потребует передачи данных, через фазу арбитража ЦУ обратится к ИУ за продолжением обмена.

С помощью сообщений согласуются параметры синхронного режима и разрядность данных. Процесс согласования синхронного обмена называется *Synchronous Negotiation*. Устройство, запрашивающее синхронный обмен, посылает сообщение *Synchronous Data Transfer Request* с указанием допустимого периода цикла и отставания REQ/ACK. Если другой участник обмена поддерживает синхронный режим, он предложит свои параметры. Согласованными параметрами будут максимальный период и минимальное отставание (нулевое отставание эквивалентно асинхронному режиму). Выбранный режим будет относиться только к фазам передачи между данной парой устройств. Отвергнутое сообщение является требованием асинхронного режима. Поскольку старые хост-адаптеры не поддерживали согласование синхронного режима, на ЦУ запрос синхронного режима может быть заблокирован. О возможности работы в синхронном режиме хост может узнать, послав команды *Request Sense* и *Inquiry*.

Разрядность передач согласуется аналогично посредством сообщений *Wide Data Transfer Request*. Согласованные режимы будут действовать до сброса устройств по сообщению *Bus Device Reset* или «жесткого» сброса, что приведет к ус-

тановке предопределенных режимов по включению. Согласование режимов не должно инициироваться в каждом процессе, поскольку затраты времени на эту процедуру сведут на нет выигрыш в производительности.

5.4. Типы ПУ

Каждое ЛУ может представлять одно или несколько однотипных *периферийных устройств (ПУ)*, перечень их стандартизованных типов приведен в табл. 5.12. Сложное ПУ может представляться несколькими ЛУ SCSI. По характеру обмена данных устройства разделяются на 2 класса — блочные (Block Device) с типами 0, 4, 5, 7 и поточные (Stream Device) с типами 1, 2, 3, 9.

Таблица 5.12. Типы ПУ SCSI

Код типа	Назначение
00h	Direct-access device — устройства прямого доступа (накопители на магнитных дисках)
01h	Sequential-access device — устройства последовательного доступа (накопители на магнитных лентах)
02h	Printer device — принтеры
03h	Processor device — процессоры (устройства обработки данных)
04h	Write-once device — устройства однократной записи (некоторые оптические диски)
05h	CD-ROM device — приводы CD-ROM
06h	Scanner device — сканеры
07h	Optical memory device — устройства оптической памяти
08h	Medium Changer device — устройства смены носителей (jukebox)
09h	Communications device — коммуникационные устройства
0Ah–0Bh	Устройства класса ASC IT8 (Graphic Arts Pre-Press Devices — высококачественные устройства печати)
0Ch	Array controller device — контроллеры массивов накопителей
0Dh–1Eh	Зарезервировано
1Fh	Неизвестный тип или устройство отсутствует

Устройства прямого доступа (0) позволяют сохранять блоки данных. Каждый блок хранится по уникальному *логическому адресу LBA — Logical Block Address.* Взаимное расположение логических блоков на носителе не регламентируется. Адрес первого логического блока — нулевой, последнего — (n−1), где n — общее число блоков. В цепочках команд устройствами может поддерживаться относительная адресация, когда исполнительный адрес в команде определяется смещением относительно адреса, действовавшего в предыдущей команде.

Блоки данных хранятся на носителе вместе с дополнительной информацией, используемой контроллером для управления чтением и записью, а также обеспечения надежности хранения данных (ECC или CRC-коды). Формат дополнительных данных не регламентируется, ЦУ скрывает эти данные от ИУ.

Для каждого блока может быть установлена своя длина, но чаще используют единую длину блока для всего носителя. Группа смежных блоков одинаковой длины называется *экстентом* (*extent*), экстенты определяются командой *MODE SELECT*, длину блока можно узнать по команде *MODE SENSE*. После изменения длины блока для активизации экстента обычно требуется форматирование.

Носитель может быть разделен на области, одна из которых используется для хранения блоков данных, другая резервируется для замены дефектных блоков, часть носителя может использоваться контроллером для обслуживания устройства. Дефектные блоки области данных могут быть переназначены на другую область носителя, что позволяет их скрыть.

Носитель может быть *фиксированным* и *сменяемым* (*Removable*). Сменяемый носитель в картридже (или чехле) называют *томом* (*Volume*). Для чтения/записи том должен быть *смонтирован.*

Устройство может быть *зарезервировано* ИУ, при этом доступ к нему других ИУ ограничивается. Ограничения распространяются на ЛУ или экстент.

Устройства, имеющие кэш данных, могут поддерживать политику обратной записи (Write Back). При этом появляют-

ся интервалы времени, в течение которых внезапное отключение питания устройства приведет к потере данных, поскольку сообщение о завершении команды посылается после записи в кэш, а не на носитель. Сообщения об ошибках при WB поступают к ИУ с опозданием. Чтобы избежать этих неудобств, ИУ может запретить устройству использовать WB. Отдельные блоки в кэше можно фиксировать, не допуская их замещения при последующих операциях обмена.

Типичный пример устройств прямого доступа — накопитель на магнитном диске. Есть устройства прямого доступа на ленточном носителе — Floppy Tape. Именно для них эффективна команда *SEEK*. Устройства прямого доступа могут не иметь подвижных носителей, а быть основаны на памяти разной природы: SRAM, DRAM, FRAM, EEPROM, флэш-память.

Устройства последовательного доступа (1) имеют ряд особенностей, связанных с принципом их действия. Носитель представляет собой магнитную ленту с многодорожечным, серпантинным или наклонно-строчным типом записи. Носитель — всегда сменяемый, с некоторым конструктивным обрамлением (катушка, картридж) — также называется *томом*. Том имеет начало носителя BOM (beginning-of-medium) и конец носителя EOM (end-of-medium). При записи ИУ должен заранее узнавать о приближении конца носителя, для чего определяется позиция раннего предупреждения EW (Early Warning) с соответствующим маркером. Это позволяет после записи блока данных из буфера поместить на носитель соответствующий концевой маркер. Том может иметь один или несколько *разделов* (Partitions), нумеруемых с нуля. Разделы располагаются друг за другом, без перекрытия. Каждый раздел x имеет начало BOP x (Begining-Of-Partition x), конец EOP x и раннее предупреждение EW x. Между началом и концом раздела помещаются блоки данных и маркеры. Блоки данных, передаваемые ИУ, называются логическими. Логический блок может занимать один или несколько физических блоков данных на носителе, в последнем случае за блокирование-деблокирование отвечает ЦУ. Описатели записанных физических и/или логических блоков могут храниться на носителе (определяется форматом записи). Буфер устройства должен вмещать по крайней мере один логический блок.

Принтеры (2), подключаемые через интерфейс SCSI, не требуют особых команд для управления, поскольку эти функции реализуются через поток передаваемых данных. Однако двунаправленная связь по шине позволяет ввести дополнительные команды, служащие для отслеживания состояния принтеров с буферной памятью, и обеспечить целостность заданий. Принтеры могут иметь встроенный контроллер SCSI или подключаться ЛУ к внешнему контроллеру по интерфейсу Centronics или RS-232. Параметры подключения определяются командой *MODE SELECT*.

Процессорными устройствами (3) в терминологии SCSI являются источники и потребители пакетов информации, трактовка которой стандартом не определена. Примерами процессорных устройств являются компьютеры, обменивающиеся сообщениями односторонним или двухсторонним образом. Процессорным устройством является и какое-либо сложное устройство отображения (графический дисплей), которое занято выводом потока сообщений. От коммуникационных устройств процессорные отличаются тем, что они являются источниками или потребителями информации, в то время как коммуникационные служат лишь посредниками.

Устройства оптической памяти (7) близки к устройствам прямого доступа со сменными носителями, но имеют ряд характерных особенностей. Большая емкость носителя вызывает необходимость применения команд с 12-байтным дескриптором. Устройства могут обеспечивать считывание, однократную или многократную запись. На носителе могут быть определены зоны, недоступные для записи. Блоки носителя имеют состояние «чистый» и «записанный», что отмечается соответствующим атрибутом. Для устройств многократной записи перед повторной записью блока необходимо его стереть. Стирание может выполняться специальной командой или автоматически по команде записи. В записи фаза стирания может отсутствовать. К этим устройствам применимо понятие *обновления* (*update*) логического блока — запись новых данных по тому же адресу логического блока, но на другое место носителя. Прежние данные могут быть считаны специальной командой, указывающей кроме логического адреса блока и его *поколение* (*generation*). Емкость носителя в таком случае сообщает-

ся без учета множества поколений. Оптические носители по сравнению с магнитными имеют существенно более высокий уровень ошибок, так что приходится использовать более сложные алгоритмы восстановления информации.

Устройства однократной записи (4), обычно оптические, отличаются невозможностью перезаписи ранее записанного блока. Попытка повторной записи в зависимости от реализации устройства может приводить к потере записанных данных. Каждый блок имеет состояние «записан» или «не записан», инициализация (форматирование) не применяется.

Приводы CD-ROM (5) предназначаются для работы с CD-дисками. Изначально диски содержали звукозапись и приводы были рассчитаны не только на чтение блоков данных, но и на потоковый вывод на внешний аудиоинтерфейс. Запись не предусмотрена. Данные на диске адресуются по-разному. Физический сектор имеет 2352 байт, из которых обычно используется 2048, поле синхронизации 12 байт и поле тега адреса сектора 4 байт. Дополнительное поле 288 байт используется для исправления ошибок данных, но если ошибки допустимы, его тоже используют для хранения данных. Таким образом, физический сектор данных может иметь размер 2048 байт (CD-ROM Data Mode 1) или 2336 и даже 2340 (вместе с полем тега) байт (CD-ROM Data Mode 2). В зависимости от размера логического блока (1024, 512 или 256 байт) сектор может вмещать 2, 4 или 8 блоков.

Один сектор, или кадр (frame), аудиодиска хранит 1/75 с звукозаписи. От этого происходит адресация MSF: 75 смежных кадров, адресуемых полем F (0–74), объединяются в более крупную единицу, адресуемую полем S (0–59), звучащую 1 с. 60 полей S соответствуют полю M (0–74), звучащему 1 мин. Адресация MSF может использоваться как абсолютная или относительно начала трека.

Носитель делится на треки (*track*), характерные однотипностью записанной информации. Каждый трек (они нумеруются от 1 до 99) делится на последовательно нумеруемые (1–99) индексы (*index*).

Носитель CD-ROM и CD-DA кроме основного канала имеет субканал (sub-channel), разделенный на 8 частей, называ-

емых P, Q, R, S, T, U, V и W. К примеру, часть Q несет информацию для контроллера и привода — управляющие поля и MSF-адрес. Каждая часть имеет производительность, равную 1/192 основного канала.

Сканеры (6) передают ИУ данные, описывающие растровое изображение сканируемого объекта. Команды позволяют задавать окна сканирования, определяя в них режим и разрешение. Для некоторых функций требуется посылка данных в сканер (например, полутоновые маски). Для сканеров с автоподачей имеются команды позиционирования.

Устройства смены носителей (8) предназначены для автоматического манипулирования сменными носителями — дисками и картриджами с магнитными лентами. В SCSI они представляются ЛУ, отличающимися от *первичных устройств* хранения, которые они обслуживают. Модель устройства состоит из набора адресуемых *элементов*, каждый из которых может «держать» только одну единицу носителя:

- Medium Transport Element — элемент транспортировки носителя;
- Storage Element — место хранения единицы носителя, когда он не находится ни в одном из элементов трех других типов;
- Import Export Element — элемент, с помощью которого устройство принимает носители извне или отдает их;
- Data Transfer Element — позиция первичного устройства, осуществляющего обмен данными с носителем.

Элементы адресуются 16-битными адресами; доступна информация об их состоянии. Каждый том носителя снабжается идентификатором-тегом, по которому определяется его нахождение в элементах.

Коммуникационные устройства (9) предназначены для обмена информацией с устройствами через внешнюю по отношению к шине SCSI среду передачи данных. Внешние протоколы стандартом SCSI не описываются: вся необходимая для них информация заключена в сообщениях, передаваемых и принимаемых ИУ по командам ***SEND MESSAGE*** и ***GET MESSAGE***.

5.5. Адресация и система команд

Как указывалось ранее, любое устройство SCSI на шине адресуется идентификатором SCSI ID, соответствующим заданному уникальному адресу. В ЦУ может быть определено до 8 ЛУ со своими номерами *LUN* (*Logical Unit Number*) в диапазоне 0–7. Понятие *LUN* неприменимо к ИУ, но SCSI-устройство двойного назначения может иметь ЛУ.

Система команд и сообщений позволяет адресовать как ЦУ в целом, так и любое его ЛУ. В ЦУ может быть определено до 8 *целевых программ TRN* (*Target Routine*), которые не имеют непосредственной привязки к ЛУ. Целевые программы появились в SCSI-2, их адресация также производится через сообщения.

Система команд SCSI включает общие команды, применимые для устройств всех классов, и специфические для каждого класса. Как общие, так и специфические наборы команд содержат обязательные (Mandatory), дополнительные (Optional) и фирменные (Vendor Specific) команды. Любое SCSI-устройство должно поддерживать обязательные команды общего набора и своего класса, чем обеспечивается высокий уровень совместимости. Команда передается ИУ в ЦУ через блок дескриптора команды *Command Descriptor Block*, посылаемый в фазе *Command*. Некоторые команды сопровождаются блоком параметров, следующим за блоком дескриптора в фазе *Data*. Форматы блоков стандартизованы, длина блока определяется кодом операции *Operation Code*, который всегда является первым байтом блока и может составлять 6, 10 или 12 байт. Типовой блок содержит следующие поля (рис. 5.8):

- *OpCode* — код операции, 1 байт (графа «Код» в табл. 5.14). Биты [7:5] определяют группу, а биты [4:0] — код команды. В группе 0 блок дескриптора имеет длину 6 байт, в группах 1 и 2 — 10 байт, в группе 5 — 12 байт. Группы 3 и 4 зарезервированы, группы 6 и 7 отданы на усмотрение разработчиков.
- *LUN* — номер ЛУ (для совместимости со SCSI-1), всегда занимает биты [7:5] байта 1. Если для идентификации используются сообщения *Identify*, то это поле игнорируется (рекомендуется устанавливать *LUN*=0).

- *LBA* — адрес логического блока, 21 бит для 6-байтных блоков и 32 бит — для 10- и 12-байтных. В ряде команд поле не используется.
- *Length* — длина (количество блоков или байт) передаваемых данных *Transfer Length*, блока параметров *Parameter List Length* или блока, резервируемого ИУ под данные *Allocation Length* (один из трех вариантов в зависимости от команды). При однобайтном задании длины 0 соответствует значению 256, в длинных формах 0 указывает на отсутствие передач. В ряде команд поле не используется.
- *Control* — байт управления. Биты [7:6] отданы на усмотрение разработчику, биты [5:2] — зарезервированы, бит 1 — *Flag*, бит 0 — *Link* (служит признаком объединения команд в цепочку). *Flag* определяет сообщение, передаваемое в случае успешного выполнения команд цепочки, при *Flag*=1 сообщение *Linked Command Complete (With Flag)* будет вызывать прерывания между командами цепочки.

а

Байт	7 6 5	4 3 2 1 0
0	Op Code	
1	LUN	(MSB)
2	LBA	
3		(LSB)
4	Length	
5	Control	

б

Байт	7 6 5	4 3 2 1 0
0	Op Code	
1	LUN	Резерв
2	(MSB)	
3	LBA	
4		
5		(LSB)
6	Резерв	
7	(MSB)	
8	Length	(LSB)
9	Control	

в

Байт	7 6 5	4 3 2 1 0
0	Op Code	
1	LUN	Резерв
2	(MSB)	
3	LBA	
4		
5		(LSB)
6	(MSB)	
7	Length	
8		
9		(LSB)
10	Резерв	
11	Control	

Рис. 5.8. Форматы блоков дескрипторов команд: *а* — 6 байт, *б* — 10 байт, *в* — 12 байт

Обратим внимание на порядок байт: первыми передаются старшие байты (бит MSB — самый старший), за ними — младшие (бит LSB — самый младший). Зарезервированные поля для совместимости с будущими стандартами должны иметь нулевые значения.

Исполнение команды завершается в фазе *Status* передачей *байта состояния Status Byte*. Байт не передается, если команда завершена по сообщению *Abort*, *Abort Tag*, *Bus Device Reset*, *Clear Queue*, по условию *Hard Reset* или в случае неожиданного разъединения. В байте состояния используются только биты [5:1], возможные состояния приведены в табл. 5.13 (остальные зарезервированы).

Таблица 5.13. Байты состояния

Биты 7 6 5 4 3 2 1 0	Состояние	Значение
R R 0 0 0 0 0 R	*Good*	Успешное завершение команды
R R 0 0 0 0 1 R	*Check Condition*	Указание на асинхронное событие
R R 0 0 0 1 0 R	*Condition Met*	Запрошенная операция выполнена (команды *Search Data* и *Pre-Fetch*)
R R 0 0 1 0 0 R	*Busy*	Занято (невозможен прием команды)
R R 0 1 0 0 0 R	*Intermediate*	Успешное выполнение команды в цепочке
R R 0 1 0 1 0 R	*Intermediate Condition Met*	Удовлетворение запрошенной операции в цепочке команд
R R 0 1 1 0 0 R	*Reservation Conflict*	Попытка обратиться к ЛУ, зарезервированному другим ИУ
R R 1 0 0 0 1 R	*Command Terminated*	Завершение текущего процесса по сообщению *Terminate I/O Process* или по асинхронному событию
R R 1 0 1 0 0 R	*Queue Full*	Очередь (маркированная) заполнена, процесс в очередь не поставлен

Набор команд для устройств классов 0–9 приведен в табл. 5.14. Любое ЦУ SCSI-2 обязано поддерживать четыре команды: *Inquiry, Request Sense, Send Diagnostic, Test Unit Ready*. Они используются для конфигурирования системы, тестирования устройств и сообщений об ошибках и исключительных ситуаций. Команда *Inquiry* позволяет получить информацию о ЛУ через стандартизованный блок данных длиной 96 байт: тип подключенного ПУ, возможность смены носителя, поддержка 32- или 16-битного расширения, синхронного обмена, относительной адресации, цепочек команд, очередей и уведомления об асинхронных событиях.

Здесь же описываются уровень поддержки стандартов SCSI (ISO, ANSI, ECMA), идентификаторы производителя, устройства и т. п. Формат блока по усмотрению производителя может быть расширен.

В графе «Применимость» указаны номера типов ПУ, соответствующие табл. 5.12 (символ * соответствует всем типам). Команды для каждого типа устройств могут быть *обязательными* — номер сопровождается символом m (Mandatory) — или *необязательными* — номер типа сопровождается символом o (Optional). Часть команд в устройствах разных классов реализуется по-разному, что отмечено обозначением *z.

Таблица 5.14. Команды SCSI

Команда	Код	Применимость	Назначение
Change Definition	40h	*o	Модификация определений операций для ЛУ
Compare	39h	*o	Побайтное сравнение данных двух ЛУ[1]
Copy	18h	*o	Копирование данных с одного ЛУ на другое[1]
Copy And Verify	3Ah	*o	Копирование данных с одного ЛУ на другое с верификацией[1]
Erase	19h	1m	Стирание (участка или до конца носителя)
Erase (10)	2Ch	7o	То же с 10-байтным блоком дескриптора
Erase (12)	ACh	7o	То же с 12-байтным блоком дескриптора
Exchange Medium	A6h	8o	Обмен носителями между двумя элементами устройства
Format	04h	2o	Выбор шрифтов и форм
Format Unit	04h	0m 7o	Форматирование устройства
Get Data Buffer Status	34h	6o	Опрос состояния буфера данных
Get Message	08h	9o	Прием пакета из коммуникационного устройства

Команда	Код	Применимость	Назначение
Get Message (10)	28h	9o	То же с 10-байтным блоком дескриптора
Get Message (12)	A8h	9o	То же с 12-байтным блоком дескриптора
Get Window	25h	6o	Получение информации о предварительно определенном окне
Initialize Element Status	07h	8o	Инициализация состояния элемента
Inquiry	12h	*m	Опрос типа устройства, уровня стандарта, идентификатора производителя, модели и т. п.
Load Unload	1Bh	1o	Загрузка/разгрузка носителя
Locate	2Bh	1o	Позиционирование на заданный логический блок
Lock-Unlock Cache	36h	0o 4o 5o 7o	Фиксация заданных логических блоков в кэше устройства и ее отмена
Log Select	4Ch	*o	Запись статистической информации, обработка которой поддерживается устройством, в ЦУ или ЛУ
Log Sense	4Dh	*o	Считывание статистической информации с ЦУ или ЛУ
Medium Scan	38h	4o 7o	Сканирование — поиск непрерывной области чистых или записанных блоков
Mode Select (6)	15h	*z	Запись параметров носителя, ЦУ или ЛУ (с 6-байтным блоком дескриптора)
Mode Select (10)	55h	*z	То же с 10-байтным блоком дескриптора
Mode Sense (6)	1Ah	*z	Считывание параметров носителя, ЦУ или ЛУ (с 6-байтным блоком дескриптора)

Команда	Код	Применимость	Назначение
Mode Sense (10)	5Ah	*z	То же с 10-байтным блоком дескриптора
Move Medium	A5h	8m	Передача носителя
Object Position	31h	6o	Позиционирование (загрузка/выгрузка) сканируемого объекта
Pause/Resume	4Bh	5o	Пауза/продолжение воспроизведения аудио
Play Audio (10)	45h	5o	Аудиовоспроизведение указанных логических блоков
Play Audio (12)	A5h	5o	То же с 12-байтным блоком дескриптора
Play Audio MSF	47h	5o	Аудиовоспроизведение с адресацией MSF
Play Audio Track/Index	48h	5o	Аудиовоспроизведение с указанием треков и индексов
Play Track Relative (10)	49h	5o	Аудиовоспроизведение с адресацией относительно трека
Play Track Relative (12)	A9h	5o	То же с 12-байтным блоком дескриптора
Position To Element	2Bh	8o	Позиционирование транспортного элемента к указанному элементу
Pre-Fetch	34h	0o 4o 5o 7o	Считывание блоков данных в кэш (без передачи ИУ)
Prevent Allow Medium Removal	1Eh	0o 1o 4o 5o 7o 8o	Запрет/разрешение смены носителя в ЛУ
Print	0Ah	2m	Печать блока данных
Read (6)	08h	0m 1m 4o 5o 7o	Чтение данных (с 6-байтным блоком дескриптора)
Read (10)	28h	0m 4m 5m 6m 7m	То же с 10-байтным блоком дескриптора
Read (12)	A8h	4o 5o 7o	То же с 12-байтным блоком дескриптора
Read Block Limits	05h	1m	Запрос ограничений на длину блока (минимальная и максимальная длины)

Команда	Код	Применимость	Назначение
Read Buffer	3Ch	*o	Чтение буфера
Read Capacity	25h	0m 4m 7m	Определение емкости ЛУ
Read Cd-Rom Capacity	25h	5m	Определение емкости CD-ROM (возможно быстрое определение с погрешностью)
Read Defect Data	37h	0o 7o	Чтение списков дефектных блоков (Plist — исходный список от изготовителя, Glist — список, заполняемый при эксплуатации)
Read Defect Data (12)	B7h	7o	То же с 12-байтным блоком дескриптора
Read Element Status	B8h	8o	Чтение состояния элементов
Read Generation	29h	7o	Чтение максимально возможного поколения для указанного логического блока
Read Header	44h	5o	Чтение заголовка логического блока CD-ROM
Read Long	3Eh	0o 4o 5o 7o	«Длинное» чтение — данные блока и поля ECC
Read Position	34h	1o	Запрос позиции данных, находящихся в буфере (адрес начала и конца, количество блоков и байт)
Read Reverse	0Fh	1o	Чтение блоков с текущей позиции в обратном направлении
Read Sub-Channel	42h	5o	Чтение данных субканала CD-ROM
Read Toc	43h	5o	Чтение таблицы содержимого CD-ROM
Read Updated Block	2Dh	7o	Чтение определенного поколения обновленного логического блока
Reassign Blocks	07h	0o 4o 7o	Переназначение дефектных блоков

Команда	Код	Применимость	Назначение
Receive	08h	3o	Прием пакета
Receive Diagnostic Results	1Ch	*o	Получение результатов диагностики
Recover Buffered Data	14h	1o 2o	Восстановление данных, посланных в буфер, но не записанных (не напечатанных) из-за ошибки
Release	17h	0m 2m 4m 5m 6m 7m 8o	Освобождение зарезервированного ЛУ, экстента или элемента
Request Sense	03h	*m	Опрос уточненного состояния
Request Volume Element Address	B5h	8o	Передача результатов команды *SEND VOLUME TAG*
Reserve	16h	0m 1m 2m 4m 5m 6m 7m 8o	Предотвращение использования ЛУ (его экстента или элемента) другим ИУ
Rewind	01h	1m	Перемотка носителя к началу раздела
Rezero Unit	01h	0o 4o 5o 7o 8o	Приведение ЛУ в определенное состояние
Scan	1Bh	6o	Сканирование данных в определенном окне
Search Data Equal	31h	0o 4o 5o 7o	Поиск данных, (не) совпадающих с эталоном[2]
Search Data Equal (12)	B1h	4o 5o 7o	То же с 12-байтным блоком дескриптора[2]
Search Data High	30h	0o 4o 5o 7o	Поиск данных, (не) больших эталона[2]
Search Data High (12)	B0h	4o 5o 7o	То же с 12-байтным блоком дескриптора[2]
Search Data Low	32h	0o 5o 7o	Поиск данных, (не) меньших эталона[2]
Search Data Low (12)	B2h	4o 5o 7o	То же с 12-байтным блоком дескриптора[2]
Seek (6)	0Bh	0o 4o 5o 7o	Позиционирование (с 6-байтным блоком дескриптора)[3]

Команда	Код	Применимость	Назначение
Seek (10)	2Bh	0o 4o 5o 7o	Позиционирование — поиск логического адреса (с 10-байтным блоком дескриптора)[3]
Send	0Ah	3m	Посылка пакета
Send (10)	2Ah	6o	Посылка данных в устройство
Send Diagnostic	1Dh	*m	Запуск теста ЦУ. Ответом будет состояние *Good*, если тест прошел успешно, или *Check Condition* в случае ошибки
Send Message	0Ah	9m	Посылка пакета в коммуникационное устройство
Send Message (10)	2Ah	9o	То же с 10-байтным блоком дескриптора
Send Message (12)	AAh	9o	То же с 12-байтным блоком дескриптора
Send Volume Tag	B6h	8o	Посылка тега тома (шаблона) для поиска его в элементах или создания нового тега
Set Limits	33h	0o 4o 5o 7o	Определение области логических адресов, над которыми могут выполняться операции цепочки команд
Set Limits (12)	B3h	4o 5o 7o	То же с 12-байтным блоком дескриптора
Set Window	24h	6m	Определение окна сканирования
Slew And Print	0Bh	2o	Прогон бумаги и печать
Space	11h	1m	Относительное позиционирование (вперед и назад) на заданное число блоков, файлов, маркеров и т. п.
Start Stop Unit	1Bh	0o 4o 5o 7o	Разрешение/запрет операций с носителем, извлечение носителя[4]
Stop Print	1Bh	2o	Останов печати с очисткой буфера или без нее

Команда	Код	Применимость	Назначение
Synchronize Buffer	10h	2o	Синхронизация буфера — печать всего содержимого, при невозможности — сообщение об ошибке
Synchronize Cache	35h	0o 4o 5o 7o	Синхронизация кэша — запись несохраненных данных заданного диапазона адресов на носитель
Test Unit Ready	00h	*m	Опрос готовности ЛУ
Update Block	3Dh	7o	Обновление логического блока
Verify	2Fh	0o 5o 7o	Верификация — проверка возможности безошибочного считывания блоков данных с носителя
Verify	13h	1o	То же
Verify (10)	2Fh	4o 5o 7o	То же или проверка чистоты блоков (с 10-байтным блоком дескриптора)
Verify (12)	AFh	4o 5o 7o	То же с 12-байтным блоком дескриптора
Write (6)	0Ah	0o 1m 4o 7o	Запись блоков данных, переданных ИУ (с 6-байтным блоком дескриптора)
Write (10)	2Ah	0o 4m 7m	Запись блоков данных, переданных ИУ (с 10-байтным блоком дескриптора)
Write (12)	AAh	4o 7o	То же с 10-байтным блоком дескриптора
Write And Verify	2Eh	0o 4o 7o	Запись блоков данных, переданных ИУ, с верификацией записи
Write And Verify (12)	AEh	7o 4o	То же с 12-байтным блоком дескриптора
Write Buffer	3Bh	*o	Запись в буфер (но не на носитель) или загрузка микрокода

Команда	Код	Применимость	Назначение
Write Filemarks	10h	1m	Запись маркера файлов
Write Long	3Fh	0o 4o 7o	«Длинная» запись — блока данных и поля ECC
Write Same	41h	0o	Запись блока данных, переданных ИУ, в группу смежных блоков или до конца носителя[5]

[1] Команды копирования и сравнения данных оперируют парой ЛУ, которые могут принадлежать как одному ЦУ, так и разным, если эту возможность поддерживает ведущее устройство копирования — *Copy Master*. Копирование возможно между устройствами любых классов. Для устройств типов 8 и 9 эти команды не применяются.

[2] Данные ищутся сравнением указанного числа логических записей с эталоном. Логические записи определяются длиной, начальным логическим блоком и смещением внутри него. Можно потребовать попадания искомых данных в один логический блок.

[3] Команда эффективна для ленточных устройств прямого доступа.

[4] Команда STOP UNIT для устройств с кэшированием перед остановом автоматически выполняет синхронизацию кэша.

[5] Первые 4 байта в записываемых блоках могут быть заменены на физический или логический адрес блока.

5.6. Выполнение команд

Мы не будем рассматривать различные ситуации, приводящие к отклонениям от нормальной последовательности событий интерфейса. К ним относятся некорректные соединения со стороны ИУ, выбор несуществующего ЛУ, неожиданные выборки ИУ, округление параметров, реакция на асинхронные события и т. п.

Рассмотрим процесс на шине SCSI на примере *одиночной команды чтения Read*. ИУ имеет активный набор указателей и несколько сохраненных наборов, по одному на каждый из допустимого числа одновременных конкурирующих процессов. ИУ восстанавливает указатели процесса в активный набор и, выиграв арбитраж, выбирает ЦУ. Как только

ЦУ выбрано, оно берет на себя управление процессом. В фазе *Selection* ИУ вводит сигнал ATN#, сообщая о намерении послать сообщение Identify с указанием адресуемого ЛУ. Таким образом устанавливается связь *I_T_L* с данным процессом и его набором указателей. ЦУ переходит в фазу *Command* и принимает блок дескриптора команды *Read*. Интерпретировав команду, ЦУ переходит в фазу *Data IN*, передает данные, затем переводится в фазу *Status* и посылает состояние *Good*. Затем в фазе *Message IN* устройство посылает сообщение *Command Complete*, после чего освобождает шину (фаза *Bus Free*). Процесс завершен.

Рассмотрим тот же пример, но *с использованием отключения от шины* (*Disconnect*) в процессе выполнения команды. Если устройство определит, что для получения затребованных данных потребуется много времени, получив команду *Read*, оно освобождает шину, послав сообщение *Disconnect*. Как только требуемые данные готовы в ЦУ, оно, выиграв арбитраж, выберет ИУ (в фазе *Reselect*) и в фазе *Message IN* пошлет ему сообщение *Identify*. По определенной этим сообщением связи *I_T_L* ИУ восстановит соответствующий набор указателей в активный и продолжит выполнение процесса, как описано выше. Если ЦУ хочет отсоединиться, когда часть данных уже передана (например, головка дошла до конца цилиндра и требуется время на позиционирование), оно посылает сообщение *Save Data Pointer*, а затем *Disconnect*. После повторного соединения передача данных возобновится с точки, определенной последним сохраненным значением указателя. Если произошла ошибка или исключение, ЦУ может повторить обмен данными, послав сообщение *Restore Pointers* или отсоединившись без сообщения *Save Data Pointers*.

Теперь рассмотрим процесс с *цепочкой связанных команд*. По успешному завершению каждой команды цепочки ЦУ автоматически переходит к исполнению следующей. Все команды цепочки адресуются к одной и той же связи *I_T_x* и являются частью одного процесса. Команды не являются полностью независимыми: при использовании бита относительной адресации последний блок, адресованный предыдущей командой, доступен для следующей. Так, например, можно исполнить команду *Search Data*, по которой на диске будет

найден блок, содержащий информацию, совпадающую с эталоном поиска. Связав с ней команду чтения *Read*, можно прочитать этот блок или блок с указанным смещением относительно найденного. По выполнении связанных команд ЦУ посылает сообщения *Linked Command Complete* (возможно, с флагом), а ИУ обновляет набор сохраненных указателей так, что они указывают на очередную команду цепочки. Команды в цепочке выполняются как одиночные, но с возможностью относительной адресации.

Команды могут исполняться с использованием *очередей*. ЦУ могут поддерживать немаркированные и маркированные очереди. Поддержка *немаркированных очередей*, определенная еще в SCSI-1, позволяет любому ЛУ (*LUN*) или целевой программе (*TRN*), занятым процессом от одного ИУ, принимать команды (начинать процесс) с другими ИУ. При этом каждый процесс идентифицируется связью *I_T_x*, где x — *LUN* или *TRN*.

Маркированные очереди (Tagged Queue) определены в SCSI-2 для ЛУ (*LUN*, но не *TRN*). Для каждой связи *I_T_L* существует своя очередь размером до 256 процессов (немаркированные очереди можно считать вырожденным случаем маркированных с одноместными очередями). Каждый процесс, использующий маркированные очереди, идентифицируется связью *I_T_L_Q*, где Q — однобайтный тег очереди (*Queue Tag*). Теги процессам назначаются ИУ, их значения на порядок выполнения операций не влияют. Повторное использование тега возможно лишь по завершении процесса с этим тегом. Постановка в очередь выполняется через механизм сообщений, при этом очередной процесс можно поставить в очередь «по-честному», а можно пропихнуть вне очереди: процесс, поставленный в очередь с сообщением *Head Of Queue Tag*, будет выполняться сразу после завершения текущего активного процесса. Процессы, поставленные в очередь с сообщением *Simple Queue Tag*, исполняются ЦУ в порядке, который оно сочтет оптимальным. Процесс, поставленный в очередь с сообщением *Ordered Queue Tag*, будет исполняться последним. ИУ может удалить процесс из очереди, сославшись на него по тегу. Изменение порядка выполнения команд ЦУ не касается порядка команд в цепочке, поскольку

цепочка принадлежит одному процессу, а в очередь ставятся именно процессы.

5.7. Конфигурирование устройств SCSI

Все устройства на шине должны быть согласованно сконфигурированы. Для них требуется программно или с помощью джамперов установить следующие основные параметры:

Идентификатор устройства SCSI ID — адрес 0–7 (для Wide SCSI 0–15), уникальный для каждого устройства на шине. Обычно хост-адаптеру, который должен иметь высший приоритет, назначается адрес 7 (15 для Wide SCSI, если все устройства 16-битные). Позиционный код, используемый для адресации, обеспечивает совместимость адресации 8- и 16-битных устройств на одной шине. Типовые заводские назначения идентификаторов устройств приведены в табл. 5.15, хотя они не являются обязательными. В настоящее время прорабатывается спецификация PnP для устройств SCSI, позволяющая автоматизировать процесс назначения идентификаторов. Она обеспечивает сосуществование традиционных (Legacy SCSI) устройств, идентификаторы которых задаются джамперами, с автоматически конфигурируемыми PnP-устройствами.

Контроль паритета — SCSI Parity. Если хотя бы одно устройство не поддерживает контроль паритета, он должен быть отключен для всех устройств на шине. Контроль паритета, особенно для дисковых устройств, является надежным средством защиты от искажения данных при передаче по шине.

Включение терминаторов — Termination. В современных устройствах применяются активные терминаторы, которые могут включаться одним джампером или программно управляемым сигналом. Терминаторы включаются только на крайних устройствах в цепочке. Современные хост-адаптеры позволяют автоматически включать свой терминатор, если они являются крайними, и отключать, если используются внутренний и внешний разъем канала. Это позволяет подключать и отключать внешние устройства, не заботясь о переключении терминаторов. Ранее приходилось открывать корпус и переставлять джампер, а пассивные терминаторы устанавливать в специальные гнезда (и извлекать их оттуда).

При отсутствии внутренних терминаторов пользователь вынужден был использовать внешние, устанавливаемые на кабель.

Правильная установка терминаторов крайне существенна — отсутствие/избыток терминаторов может привести к неустойчивости или неработоспособности интерфейса.

Питание терминаторов (Terminator Power). Когда используются активные терминаторы (для современных устройств — всегда), питание терминаторов должно быть включено (джампером или программно) хотя бы на одном устройстве.

Согласование скорости синхронного обмена (SCSI Synchronous Negotiation). Режим синхронного обмена, обеспечивающий высокую производительность, включается по взаимному согласию устройств. Если хотя бы одно устройство на шине его не поддерживает, рекомендуют согласование запретить на хост-адаптере. Если обмен будет инициирован синхронным устройством, хост-адаптер поддержит этот режим.

Старт по команде (Start on Command) или задержанный старт (Delayed Start). При включении этой опции запуск двигателя устройства выполняется только по команде от хост-адаптера, что позволяет снизить пик нагрузки блока питания в момент включения. Хост будет запускать устройства последовательно.

Разрешение отключения (Enable Disconnection). Позволяет устройствам отключаться от шины при неготовности данных во время длительных операций с носителем, что весьма эффективно в многозадачном режиме при нескольких ПУ на шине. В случае одного устройства отключение приводит только к дополнительным затратам времени на повторное соединение.

Таблица 5.15. Заводская установка идентификаторов устройств

SCSI ID	Устройство
7	Хост-адаптер
6	Накопитель на магнитных дисках
5	–
4	Ленточный или R/W-оптический накопитель
3	CD-ROM

SCSI ID	Устройство
2	Сканер, принтер
1	НЖМД, поддерживаемый BIOS хост-адаптера
0	НЖМД, поддерживаемый BIOS хост-адаптера

Устройства SCSI допускают программирование — *Programmable Operating Definition*. Программированием определений операций для ЛУ можно изменить такие параметры, как идентификация производителя, типа и модели устройства, уровень соответствия SCSI, номер спецификации, набор команд и т. д. Однако низкоуровневые параметры интерфейса (параметры временных диаграмм, определение паритета) сохраняются неизменными. Текущие определения могут быть считаны ИУ по командам *Inquiry*, *Mode Sense* и *Read Capacity*.

5.8. Хост-адаптер SCSI

Хост-адаптер является важнейшим узлом интерфейса, определяющим производительность системы SCSI. Существует широкий спектр адаптеров. К простейшим можно подключать только устройства, не критичные к производительности. Такие адаптеры входят в комплект поставки сканеров, а подключение к ним диска невозможно. Высокопроизводительные адаптеры имеют собственный специализированный процессор, большой объем буферной памяти и используют высокоэффективные режимы прямого управления шиной для доступа к памяти компьютера. Адаптеры SCSI существуют для всех шин: ISA (8–16 бит), EISA, MCA, PCI, VLB, PCMCIA и для параллельного порта. Ряд системных плат имеют встроенный SCSI-адаптер, подключенный к одной из локальных шин. При выборе интерфейса, к которому подключается хост-адаптер, учитывайте производительность — интерфейс не должен стать узким местом при обмене с высокопроизводительными устройствами SCSI. Наибольшую эффективность имеют хост-адаптеры для шины PCI. Конечно, за мощный адаптер для сервера приходится платить — его цена может превышать цену рядового настольного компьютера. Еще дороже хост-адаптеры со встроенными контроллерами RAID-массивов, которые содержат мощный RISC-процессор и большой объем локальной памяти.

Конфигурирование хост-адаптеров с точки зрения шины SCSI не отличается от конфигурирования других устройств. Для современных адаптеров вместо джамперов используется программное конфигурирование. Утилита конфигурирования обычно входит в расширение BIOS, установленное на плате адаптера, и приглашение к ее вызову выводится на экран во время POST.

Как и всякая карта расширения, хост-адаптер должен быть сконфигурирован и с точки зрения шины расширения, к которой он подключается. Системные ресурсы для шинного SCSI-адаптера включают:

- область памяти для расширения ROM BIOS, необходимого для поддержки конфигурирования устройств и дисковых функций. Если в системе установлено несколько однотипных хост-адаптеров, ROM BIOS для них используется с одного адаптера. Разнотипные хост-адаптеры не всегда могут работать вместе;
- область разделяемой буферной памяти;
- область портов ввода-вывода (I/O Port);
- IRQ — запрос прерывания;
- DMA — канал прямого доступа к памяти (для шин ISA/EISA), часто используемый для захвата управления шиной (Bus-Mastering).

Все устройства SCSI требуют специальных драйверов. Базовый драйвер дисковых устройств входит в BIOS хост-адаптера. Расширения, например ASPI (Advanced SCSI Programming Interface), загружаются отдельно. От драйверов сильно зависит производительность устройств SCSI. «Умное» ПО способно эффективно загружать работой устройства, а иногда и «срезать углы» — выполнять копирование данных между устройствами без выхода на системную шину компьютера.

6. Интерфейс ATA

Интерфейс *ATA— AT Attachment for Disk Drives* — разрабатывался в 1986–1990 гг. для подключения накопителей на жестких магнитных дисках к компьютерам IBM PC AT с шиной ISA. Стандарт, выработанный комитетом X3T10, определяет набор регистров и назначение сигналов 40-контактного интерфейсного разъема. Интерфейс появился в результате переноса контроллера жесткого диска ближе к накопителю, то есть создания устройств со встроенным контроллером — *IDE* (*Integrated Device Electronic*). Стандартный для AT контроллер жесткого диска был перенесен на плату электроники накопителя с сохранением регистровой модели. При этом удлинилась связь с устройством со стороны системной шины, которую непосредственно выводить на длинный ленточный кабель было нецелесообразно. Это сказалось бы на скорости работы шины, надежности и цене. Из всех сигналов шины ISA выбрали минимальный набор сигналов, часть из которых буферизовали на небольшой плате, устанавливаемой в слот, а часть направили прямо на разъем ленточного кабеля нового интерфейса. Из сигналов системной шины потребовались следующие:

- шина данных — используется полностью;
- шина адреса — 3 младших бита поступают в интерфейс, старшие биты и сигнал AEN проходят через дешифратор, вырабатывающий сигналы выбора устройства CS0# и CS1#;
- шина управления — используются сигналы чтения и записи портов и аппаратного сброса, запрос прерывания, пара сигналов канала DMA, сигналы готовности и управления разрядностью передачи.

Поскольку стандартный контроллер AT позволял подключать до двух накопителей, эту возможность получил и интерфейс ATA. Однако теперь два накопителя стали означать и два контроллера. Их подключили к одной интерфейсной

шине, а для программной совместимости бит выбора накопителя (*DEV*) в регистре номера головки и номера устройства (*Drive/Head register*) стали использовать для выбора устройства. Для взаимодействия пары устройств на шине ввели несколько дополнительных сигналов. Так появился интерфейс *ATA* для подключения устройств IDE к шине ISA. Позже их стали подключать и к локальным шинам, но набор сигналов интерфейса и протоколы обмена сохранились. Достаточно универсальный набор сигналов позволяет подключать любое устройство со встроенным контроллером, которому в пространстве портов ввода/вывода достаточно того же набора регистров и которое способно поддержать режим выбора устройства через вышеупомянутый бит. Принятая система команд и регистров, являющаяся частью спецификации ATA, ориентирована на блочный обмен данными с устройствами прямого доступа. Для иных устройств существует спецификация *ATAPI*, основанная на тех же аппаратных средствах, но позволяющая обмениваться пакетами управляющей информации (*PI — Package Interface*). Пакетный интерфейс позволяет расширить границы применения шины ATA.

Адресация в ATA тоже имеет «дисковые корни»: для накопителей изначально указывали адрес цилиндра (Cylinder), головки (Head) и сектора (Sector) — так называемая *трехмерная адресация CHS*. Позже по ряду причин стали различать *физическую* (реальную для накопителя) и *логическую* (по которой с устройством общается программа) *адресацию CHS*. При этом одно и то же устройство могло иметь различную *логическую геометрию* (но, естественно, $C \times H \times S_{лог} \leq C \times H \times S_{физ}$). Преобразование логической адресации в физическую выполняется встроенным контроллером устройства. Позже пришли к *линейной адресации логических блоков LBA* (Logical Block Addressing), где адрес блока (сектора) определяется 28-битным числом.

В спецификации ATA фигурируют следующие компоненты:

- *Хост-адаптер* — средства сопряжения интерфейса ATA с системной шиной (набор буферных схем между шинами ISA и ATA). *Хостом* мы будем называть компьютер с хост-адаптером интерфейса ATA.

- *Кабель-шлейф* с двумя или тремя 40-контактными IDC-разъемами. В стандартном кабеле одноименные контакты всех разъемов соединяются вместе.
- *Ведущее устройство* (Master) — ПУ, в спецификации ATA официально называемое *Device-0* (устройство 0).
- *Ведомое устройство* (Slave) — ПУ, в спецификации официально называемое *Device-1* (устройство 1).

Если к шине ATA подключено одно устройство, оно должно быть ведущим. Если подключены два устройства, одно должно быть ведущим, другое — ведомым. О своей роли (ведущее или ведомое) устройства «узнают» с помощью предварительно установленных конфигурационных джамперов. Если применяется «кабельная выборка» (см. ниже), роль устройства определяется его положением на специальном кабеле-шлейфе.

Оба устройства воспринимают команды от хост-адаптера одновременно. Однако исполнять команду будет лишь *выбранное* устройство. Если бит *DEV*=0, выбрано ведущее устройство, если *DEV*=1 — ведомое. Выходные сигналы на шину ATA имеет право выводить только выбранное устройство. Такая система подразумевает, что, *начав операцию обмена с одним из устройств, хост-адаптер не может переключиться на обслуживание другого* до завершения начатой операции. Параллельно могут работать только устройства IDE, подключаемые к разным шинам (каналам) ATA. Спецификация ATA-4 определяет способ обхода этого ограничения (см. 6.6).

Выполняемая операция и направление обмена данными между устройством и хост-адаптером определяются предварительно записанной *командой*. Непременным компонентом устройства является *буферная память*. Ее наличие позволяет выполнять обмен данными в темпе, предлагаемом хост-адаптером (в пределах возможности устройства), без оглядки на внутреннюю скорость передачи данных между носителем и буферной памятью ПУ.

Для подключения устройств IDE существует несколько *разновидностей интерфейса*:

- ATA, он же AT-BUS — 16-битный интерфейс подключения к шине компьютера AT. Наиболее распространенный

40-проводной сигнальный и 4-проводной питающий интерфейс для подключения дисковых накопителей к компьютерам AT. Для миниатюрных (2,5" и менее) накопителей используют 44-проводной кабель, по которому передается и питание.

- PC Card ATA — 16-битный интерфейс с 68-контактным разъемом PC Card (PCMCIA) для подключения к блокнотным PC.
- XT IDE (8-бит), он же XT-BUS — 40-проводный интерфейс, похожий на ATA, но несовместимый с ним.
- MCA IDE (16-бит) — 72-проводный интерфейс, предназначенный специально для шины и накопителей PS/2.
- *ATA-2* — расширенная спецификация ATA. Включает 2 канала, 4 устройства, PIO Mode 3, multiword DMA mode 1, Block mode, объем диска до 8 Гбайт, поддержка LBA и CHS.
- *Fast ATA-2* разрешает использовать Multiword DMA Mode 2 (13,3 Мбайт/с), PIO Mode 4.
- *ATA-3* — расширение ATA-2. Включает средства парольной защиты, улучшенного управления питанием, самотестирования с предупреждением приближения отказа — SMART (Self Monitoring Analysis and Report Technology).
- *ATA/ATAPI-4* — расширение ATA-3, включающее режим *Ultra DMA* со скоростью обмена до 33 Мбайт/с и пакетный интерфейс ATAPI.
- *E-IDE* (*Enhanced IDE*) — расширенный интерфейс, введенный фирмой Western Digital. Реализуется в адаптерах для шин PCI и VLB. Позволяет подключать до 4 устройств (к двум каналам), включая CD-ROM и стриммеры (ATAPI). Поддерживает PIO Mode 3, multiword DMA mode 1, объем диска до 8 Гбайт, LBA и CHS. С аппаратной точки зрения практически полностью соответствует спецификации ATA-2.

Устройства ATA IDE, E-IDE, ATA-2, Fast ATA-2, ATA-3 и ATA/ATAPI-4 электрически совместимы. Степень логической совместимости достаточно высока (все базовые возможности ATA доступны). Однако для полного использования всех расширений необходимо соответствие спецификаций устройств, хост-адаптера и его ПО. В настоящее время наиболее широко

распространен и четко стандартизован интерфейс, официально называемый *ATA-2*, о котором в основном и пойдет речь.

6.1. Электрический интерфейс

Все информационные сигналы интерфейса передаются через 40-контактный разъем, у которого ключом является отсутствующий на вилке и закрытый на розетке контакт № 20. Использование в качестве ключа выступа на корпусе розетки и прорези в бандаже вилки стандартом не приветствуется. Для соединения устройств спецификация требует применения плоского многожильного кабеля (Flat cable stranded 28 AWG) типа 3M 3365-40 или эквивалентного. Возможно применение экранированного кабеля (Flat cable stranded 28 AWG 3M 3517-40 shielded). Длина кабеля не должна превышать 0,46 м (18"), допустимая емкость проводников не более 35 пФ. Терминаторы стандартом не предусматриваются (они имеются в каждом устройстве и хост-адаптере), но если кабель с тремя разъемами (розетками) используют для подключения одного устройства, то и его, и хост-адаптер рекомендуется подключать к противоположным концам кабеля. Состав информационных сигналов интерфейса ATA приведен в табл. 6.1. Все сигналы ATA являются логическими со стандартными ТТЛ-уровнями. Вид кабеля приведен на рис. 6.1. Одноименные контакты всех разъемов соединяются своими проводами. Встречается (редко) шлейф с кабельной выборкой (рис. 6.2). В нем провод 28 в шлейфе перерезан, так что контакт 28 (*CSEL*) для ведущего устройства заземлен через хост-адаптер, а для ведомого — не подключен. Кабель должен соответствовать системе адресации, выбранной для обоих устройств. Для устойчивой работы в режиме Ultra DMA рекомендуется применение 80-проводных кабелей, обеспечивающих чередование сигнальных цепей и проводов схемной земли (эти кабели «разделываются» на обычные 40-контактные разъемы). Устройства и адаптеры, рассчитанные на использование Ultra DMA, должны иметь в сигнальных цепях последовательные согласующие резисторы с сопротивлением для различных цепей 22, 33 или 82 Ом. Спецификация ATA «узаконивает» 40-контактный сигнальный и 4-контактный разъемы питания (рис. 6.3б), но для малогабаритных устройств питание может подаваться по 44-проводному интерфейсному кабелю.

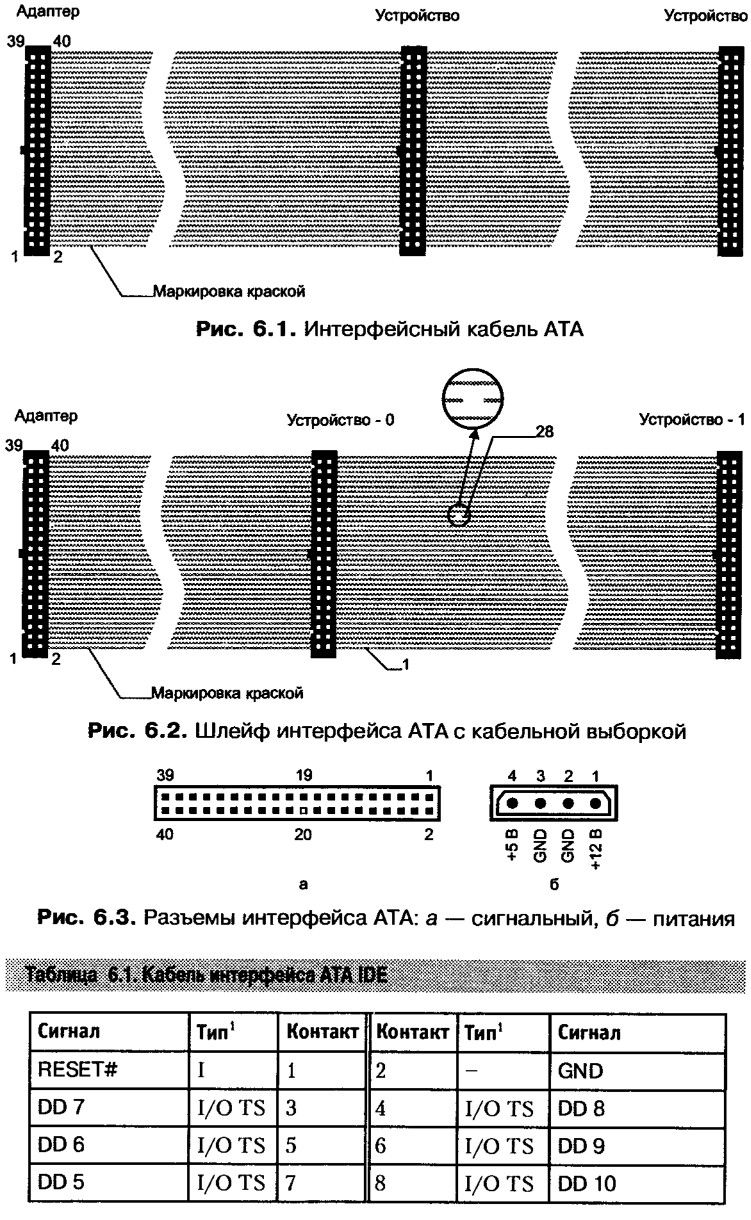


Рис. 6.1. Интерфейсный кабель ATA

Рис. 6.2. Шлейф интерфейса ATA с кабельной выборкой

Рис. 6.3. Разъемы интерфейса ATA: *а* — сигнальный, *б* — питания

Таблица 6.1. Кабель интерфейса ATA IDE

Сигнал	Тип[1]	Контакт	Контакт	Тип[1]	Сигнал
RESET#	I	1	2	–	GND
DD 7	I/O TS	3	4	I/O TS	DD 8
DD 6	I/O TS	5	6	I/O TS	DD 9
DD 5	I/O TS	7	8	I/O TS	DD 10

Сигнал	Тип[1]	Контакт	Контакт	Тип[1]	Сигнал
DD 4	I/O TS	9	10	I/O TS	DD 11
DD 3	I/O TS	11	12	I/O TS	DD 12
DD 2	I/O TS	13	14	I/O TS	DD 13
DD 1	I/O TS	15	16	I/O TS	DD 14
DD 0	I/O TS	17	18	I/O TS	DD 15
GND	–	19	20	–	Ключ (нет штырька)
DMARQ	O TS[2]	21	22	–	GND
DIOW# /STOP [3]	I	23	24	–	GND
DIOR# /HDMARDY# /HSTROBE [3]	I	25	26	–	GND
IORDY /DDMARDY# /DSTROBE [3]	O TS[2]	27	28	I/O	SPSYNC/CSEL
DMACK#	I	29	30	–	GND
INTRQ	O TS[2]	31	32	O OK	IOCS16#
DA 1	I	33	34	I, O[4]	PDIAG# /CBLID [3]
DA 0	I	35	36	I	DA 2
CS0#	I	37	38	I	CS1#
DASP#	I/O OK[5]	39	40	–	GND
+5 В (Logic)	–	41[6]	42[6]	–	+5 В (Motor)
GND	–	43[6]	44[6]	–	Зарезервирован

[1] Тип сигнала для устройства: I — вход, O — выход, I/O — двунаправленный, TS — тристабильный, OC — открытый коллектор.

[2] У старых устройств сигнал может иметь тип OC (при разнотипных сигналах на одной шине возможен конфликт).

[3] Сигналы, приведенные после символа /, используются только в режиме Ultra DMA (ATA-4).

[4] У ведущего устройства — вход, у ведомого — выход.

[5] У ведомого устройства — только выход.

[6] Контакты 41–44 используются только для миниатюрных дисков.

В документации на устройства могут применяться несколько отличающиеся обозначения сигналов. Здесь приведены обозначения из стандарта ATA/ATAPI-4. Сигналы имеют следующее назначение:

RESET# (Device reset) — сброс устройства (инвертированный сигнал сброса системной шины). Сигнал длительностью не менее 25 мкс вырабатывается после установления питающих напряжений.

DA[2:0] (Device Address) — три младших бита системной шины адреса, используемые для выбора регистров устройств.

DD[15:0] (Device Data) — двунаправленная 16-битная шина данных между адаптером и устройствами. При 8-битных обменах используются младшие биты D[7:0].

DIOR# (Device I/O Read) — строб чтения портов ввода/вывода. Данные фиксируются по положительному перепаду сигнала.

DIOW# (Device I/O Write) — строб записи портов ввода/вывода. Данные фиксируются по положительному перепаду сигнала.

IORDY (I/O channel ready) — готовность устройства завершить цикл обмена. Низким уровнем сигнала во время цикла обмена устройство может ввести такты ожидания шины. Сигнал требуется при обмене в PIO MODE 3 и выше.

IOCS16# — разрешение 16-битных операций. Обращение ко всем регистрам, кроме регистра данных, всегда 8-битное. Для PIO MODE 0, 1, 2 при активном сигнале обращения 16-битные, при неактивном — 8-битные. Для PIO MODE 3, 4 и DMA все обмены 16-битные, кроме дополнительных байт (выходящих за границу 512-байтного сектора) «длинного» считывания и записи.

DMARQ (DMA ReQuest) — запрос обмена по каналу DMA (сигнал необязательный). При разрешении обмена сигнал (высокий уровень) вводится устройством по готовности к обмену. Введя сигнал DMARQ, устройство должно дождаться подтверждения от хост-адаптера сигналом DMACK#, после чего может снять запрос DMARQ. Для очередной передачи запрос должен быть введен снова. В режиме *Multi-Word DMA* запрос может

удерживаться на время передачи всех данных. Выход должен быть тристабильным, в активном состоянии (0 или 1) он может быть только у выбранного устройства во время работы с DMA. В ATA-1 для этого сигнала мог использоваться как тристабильный, так и стандартный ТТЛ-выход. Работа на одной шине устройств с разнотипными выходами DMARQ может привести к конфликтам.

DMACK# (DMA aCKnowledge) — подтверждение DMA. Сигнал вырабатывается хост-адаптером как подтверждение цикла передачи. Передача слова данных управляется сигналами DIOR# или DIOW#. Во время обмена по каналу DMA сигналы IOCS16#, CS0# и CS1# не используются, обмен всегда производится 16-битными словами.

INTRQ (Device interrupt) — запрос прерывания. Выход должен быть тристабильным, активный сигнал (логическую 1) вырабатывает только выбранное устройство, когда у него имеется необслуженный запрос прерывания и его вырабатывание не запрещено битом *nIEN* в регистре *Device Control*. Запрос сбрасывается по сигналу RESET#, установке бита *SRST* в регистре *Device Control*, записи в регистр команд или чтении регистра состояния. При обменах PIO запрос устанавливается в начале передачи каждого блока (сектора или группы секторов при многосекторных операциях). Исключения: по командам *Format Track*, *Write Sector(S)*, *Write Buffer* и *Write Long* в начале передачи первого блока данных запрос прерывания не вырабатывается. При обменах DMA запрос прерывания вырабатывается только по завершении операции.

CS0# (Chip Select 0) — сигнал выбора блока командных регистров (*Command Block Registers*). Для первого канала он вырабатывается при наличии на системной шине адреса порта ввода/вывода в диапазоне 1F0h–1F7h (сигнал также называют CS1FX#).

CS1# (Chip Select 1) — выбор блока управляющих регистров (*Control Block Registers*). Для первого канала он вырабатывается при наличии на системной шине адреса порта ввода/вывода в диапазоне 3F6h–3F7h (часто этот сигнал называется CS3FX#).

PDIAG# (Passed diagnostics) — сигнал о прохождении диагностики. Ведущее устройство наблюдает за этим сигналом, который ведомое устройство должно выработать в ответ на сброс или команду диагностики. Если ведомое устройство обнаружено (по сигналу DASP#), ведущее устройство ожидает сигнал в течение 31 с после сброса и 6 с после команды диагностики. Если за это время сигнал не появился, ведущее устройство отмечает этот факт установкой бита 7 регистра ошибок. Если ведомое устройство не обнаружено, ведущее обнуляет регистр состояния ведомого устройства и сообщает свое состояние сразу после завершения собственной самодиагностики. Сигнал служит только для связи двух устройств и хост-адаптером не используется (в АТА-4 контакт задействован для сигнала CBLID#).

CBLID# (Cable assembly type identifier) — идентификация типа кабеля. В 80-проводной сборке контакт 34 на разъеме хост-адаптера соединяется с шиной GND, а контакты 34 разъемов устройств соединяются между собой, но связи с разъемом хост-адаптера не имеют. После прохождения сброса (когда сигнал PDIAG# снимается) хост может определить наличие 80-проводного кабеля по низкому уровню сигнала.

DASP# (Device Active, Slave Present) — сигнал двойного назначения: индикатор активности устройства и присутствия ведомого устройства. Устройства имеют выход типа «открытый коллектор» с нагрузочным резистором 10 кОм к шине +5 В. После сброса по сигналу RESET# или при инициализации по включении питания оба устройства в течение 1 мс должны деактивировать этот сигнал, после чего не позже чем через 400 мс его вводит ведомое устройство для сообщения о своем присутствии. Ведущее устройство не активирует этот сигнал в течение 450 мс. Сигнал деактивируется ведомым устройством после получения им команды или через 31 с автоматически (смотря что произойдет раньше). После этого сигнал может быть введен любым устройством как индикатор активности. Адаптер использует этот сигнал для включения светодиодного индикатора доступа к диску.

SPSYNC/CSEL (Spindle Synchronization/Cable Select) — синхронизация шпинделя/выборка кабелем. Сигнал двойного

назначения, которое должно быть единым для обоих устройств. Сигнал SPSYNC позволяет синхронизировать шпиндели устройств (актуально для RAID-массивов); используется по усмотрению производителя накопителя. Сигнал CSEL позволяет устройствам определять свой адрес по положению на специальном кабеле с разрывом провода 28 между разъемами двух устройств (малораспространенная «кабельная выборка»). Эта линия на хост-адаптере заземлена, и ведущее устройство получает заземленную линию, а ведомое — не подключенную. Состояние сигнала (если он управляется хост-адаптером) должно удерживаться по крайней мере 31 с после сигнала RESET#.

При использовании режима *Ultra DMA* четыре линии получают новое назначение сигналов:

- STOP (Stop Ultra DMA burst) — останов передачи пакета Ultra DMA.
- DDMARDY# (Device Ultra DMA ready) — готовность устройства при приеме пакета Ultra DMA (управление потоком).
- DSTROBE (Host Ultra DMA data strobe) — строб данных устройства при передаче пакета хосту. Данные передаются по обоим перепадам DSTROBE.
- HDMARDY# (Host Ultra DMA ready) — готовность хоста при приеме им пакета Ultra DMA (управление потоком).
- HSTROBE (Host Ultra DMA data strobe) — строб данных хоста при передаче пакета устройству. Данные передаются по обоим перепадам HSTROBE.

Для компьютеров класса XT существует 8-битная версия интерфейса, называемая *XT-IDE*, реже — *XT-Bus*. Этот интерфейс (табл. 6.2), как и ATA, реализован на 40-проводном кабеле, и многие его сигналы совпадают с 16-битной шиной ATA. Интерфейс XT-IDE можно рассматривать как подмножество ATA, хотя прямой совместимости нет. Ряд устройств ATA имеют джампер выбора режима XT/AT (в накопителях Seagate на это указывает окончание AX в обозначении модели).

Таблица 6.2. Кабель интерфейса XT IDE

Сигнал	Тип[1]	Контакт	Контакт	Сигнал
RESET#	I	1	2	GND
DD 7	I/O TS	3	4	GND
DD 6	I/O TS	5	6	GND
DD 5	I/O TS	7	8	GND
DD 4	I/O TS	9	10	GND
DD 3	I/O TS	11	12	GND
DD 2	I/O TS	13	14	GND
DD 1	I/O TS	15	16	GND
DD 0	I/O TS	17	18	GND
GND	–	19	20	Ключ (нет штырька)
AEN	I	21	22	GND
DIOW#	I	23	24	GND
DIOR#	I	25	26	GND
DMACK#	I	27	28	GND
DMARQ	O OC	29	30	GND
INTRQ	O TS[2]	31	32	GND
DA 1	I	33	34	GND
DA 0	I	35	36	GND
CS0#	I	37	38	GND
DASP#	I/O OC[3]	39	40	GND

[1] Тип сигнала для устройства: I — вход, O — выход, I/O — двунаправленный, TS — тристабильный, OC — открытый коллектор.

[2] У старых устройств сигнал может иметь тип OC (при разнотипных сигналах на одной шине возможен конфликт).

[3] У ведомого устройства — только выход.

6.2. Регистры устройств

Каждое устройство ATA имеет *стандартный набор регистров*, адресуемых сигналами от хост-адаптера (CS0#, CS1#, DA2, DA1, DA0, DIOR# и DIOW#). Набор регистров (табл. 6.3)

состоит из двух блоков, выбираемых сигналом CS0#/CS1#. *Блок командных регистров* служит для посылки команд устройству и чтения информации о его состоянии. *Блок управляющих регистров* используется для управления устройством и получения более подробной информации о его состоянии. На действительность содержимого регистров командного блока и альтернативного регистра состояния указывает нулевое значение бита *BSY* регистра состояния. Если устройство поддерживает управление энергопотреблением, в «спящем» режиме содержимое этих регистров недействительно.

Таблица 6.3. Регистры контроллеров устройств ATA

Адрес					Назначение (R — чтение, W — запись)
CS0#	**CS1#**	**DA2**	**DA1**	**DA0**	
1	1	x	x	x	Не используется (шина данных в третьем состоянии)
Control Block Registers — блок управляющих регистров					
1	0	0	x	x	Не используется (шина данных в третьем состоянии)
1	0	1	0	x	Не используется (шина данных в третьем состоянии)
1	0	1	1	0	R: *Alternate Status* (*AS*) — альтернативный регистр состояния W: *Device Control* (*DC*) — регистр управления устройством
1	0	1	1	1	Не используется[1]
Command Block Registers — блок командных регистров					
0	1	0	0	0	R/W: *Data* (*DR*) — регистр данных
0	1	0	0	1	R: *Error* (*ER*) — регистр ошибок W: *Features* (*FR*) — регистр свойств
0	1	0	1	0	R/W: *Sector Count* (*SC*) — регистр счетчика секторов
0	1	0	1	1	R/W: *Sector Number* (*SN*) — регистр номера сектора LBA [7:0][2]

Таблица 6.3 (*продолжение*)

Адрес					Назначение (R — чтение, W — запись)
CS0#	CS1#	DA2	DA1	DA0	
Command Block Registers — блок командных регистров					
0	1	1	0	0	R/W: *Cylinder Low* (*CL*) — регистр младшего байта номера цилиндра LBA[15:8][2]
0	1	1	0	1	R/W: *Cylinder High* (*CH*) — регистр старшего байта номера цилиндра LBA[23:16]2
0	1	1	1	0	R/W: *Device/Head* (*DH*) — регистр номера устройства и головки LBA[27:24][2]
0	1	1	1	1	R: *Status* (*SR*) — регистр состояния W: *Command* (*CR*) — регистр команд
0	0	x	x	x	Недопустимый адрес

[1] Регистр выпадает из блока. Рекомендуется, чтобы на сигнал чтения по этому адресу устройство не отвечало. Если устройство отвечает, то оно не должно управлять битом DD7 во избежание конфликта с контроллером НГМД.

[2] Регистры сектора, цилиндра и головки в режиме LBA содержат указанные биты логического адреса.

Регистр состояния *SR* отражает текущее состояние устройства в процессе выполнения команд. Чтение регистра состояния разрешает дальнейшее изменение его бит и сбрасывает запрос аппаратного прерывания.

Назначение бит регистра *SR*:

- Бит 7 — ***BSY(Busy)*** указывает на занятость устройства. При ***BSY=1*** устройство игнорирует попытки записи в командный блок регистров, а чтение этих регистров дает неопределенный результат. При ***BSY=0*** регистры командного блока доступны, а устройство может изменять только значение бит *IDX*, *DRDY*, *DF*, *DSC* и *CORR*. Бит устанавливается при аппаратном/программном сбросе устройства и по получении команды. Бит может устанавливаться на кратковременный интервал, так что хост может этого не заметить.

- Бит 6 — *DRDY (Device Ready)* указывает на готовность устройства к восприятию любых кодов команд. Если состояние бита изменилось, оно не может измениться обратно до чтения регистра состояния. При *DRDY=0* устройство воспринимает только команды *Execute Device Diagnostic* и *Initialize Device Parameters*, прекращая выполнение текущей команды и сообщая об этом флагом *ABRT* в регистре ошибок и *ERR* в регистре состояния. Другие команды приводят к непредсказуемым результатам.
- Бит 5 — *DF (Device Fault)* — индикатор отказа устройства.
- Бит 4 — *DSC (Device Seek Complete)* — индикатор завершения поиска трека. В командах, допускающих перекрытие (см. 6.6), бит называется *SERV (Service Required)* — устройство требует обслуживания.
- Бит 3 — *DRQ (Data Request)* — индикатор готовности к обмену словом или байтом данных.
- Бит 2 — *CORR (Corrected Data)* — индикатор исправленной ошибки данных.
- Бит 1 — *IDX (InDeX)* — индекс, трактуется особо каждым производителем.
- Бит 0 — *ERR (Error)* — индикатор ошибки выполнения предыдущей операции. Дополнительная информация содержится в регистре ошибок. Если установлен бит *ERR*, до приема следующей команды, программного или аппаратного сброса устройство не изменит состояние этого бита, а также регистра ошибок, регистра количества секторов и регистров цилиндра, головки и номера сектора.

В стандарте АТА-4 для некоторых команд биты 4 и 5 могут иметь иное назначение, а биты 1 и 2 объявлены устаревшими.

Альтернативный регистр состояния AS имеет те же биты, что и основной, но его чтение не приводит ни к каким изменениям состояния устройства.

Назначение *регистра команд CR* очевидно из названия. Устройство начинает исполнять команду сразу, как только ее код записан в данный регистр.

Регистры номера цилиндра (старшего *CH* и младшего байта *CL*) и *номера сектора SN* имеют двоякое назначение в зависимости от выбранной системы адресации (CHS или LBA).

Они инициализируются хост-адаптером, а в случае возникновения ошибки при операции устройство поместит в них адрес, по которому встретилась ошибка.

Регистр номера устройства и головки *DH* кроме хранения части адресной информации служит для выбора ведущего или ведомого устройства и метода адресации.

Назначение бит регистра *DH*:

- Биты 7 и 5 зарезервированы.
- Бит 6 — *L* — единичным значением указывает на применение режима адресации LBA. При нулевом значении бита используется режим CHS.
- Бит 4 — *DEV* (*Device*) — выбор устройства. При *DEV*=0 выбрано ведущее, при *DEV*=1 — ведомое.
- Биты [3:0] имеют двоякое назначение в зависимости от выбранной системы адресации. В режиме CHS они содержат номер головки, в режиме LBA — старшие биты логического адреса. Как и предыдущие, этот адресный регистр инициализируется хост-адаптером, а в случае возникновения ошибки при операции устройство поместит в них адрес, по которому встретилась ошибка.

До принятия ATA-2 считалось, что адресные регистры должны модифицироваться и после успешного выполнения операции, отражая текущее значение адреса в носителе.

После включения питания, сброса или выполнения диагностики устройство в блоке командных регистров содержит *сигнатуру*, определяющую его тип:

- *SC*=01h, *SN*=01h, *CL*=00h, *CH*=00h, *DH*=00h — устройства ATA;
- *SC*=01h, *SN*=01h, *CL*=14h, *CH*=EBh, *DH*=00h или 01h — устройства ATAPI.

Регистр данных *DR* может использоваться как 8-битный или 16-битный в зависимости от типа данных, передаваемых в текущей команде.

Из *регистра управления устройством* *CR* используются только два бита.

Назначение бит регистра *CR*:

- Биты [7:3] зарезервированы.

- Бит 2 — *SRST (Software Reset)* — программный сброс, действует все время, пока бит не будет снят. Оба устройства на шине воспринимают программный сброс одновременно. На устройства ATAPI не действует.
- Бит 1 — *nIEN (Interrupt Enable)* — инверсный бит разрешения прерывания. При нулевом значении бита выбранное устройство может вырабатывать сигнал INTRQ через тристабильный выход.
- Бит 0 всегда должен быть нулевым.

Регистр ошибок ER хранит состояние выполнения последней операции или диагностический код. После завершения операции на наличие ошибки указывает бит *ERR* регистра состояния.

Назначение бит регистра *ER*:

- Бит 7 зарезервирован.
- Бит 6 — *UNC (Uncorrectable Data Error)* — неисправимая ошибка данных.
- Бит 5 — *MC (Media Changed)* — смена носителя. После смены носителя первая команда обращения отвергается и устанавливается данный бит. После сброса бита следующие команды будут выполняться нормальным образом.
- Бит 4 — *IDNF (ID Not Found)* — указывает на ненайденный идентификатор сектора.
- Бит 3 — *ABRT (Aborted Command)* — устанавливается, если команда отвергнута как недействительная или в случае возникновения иной ошибки.
- Бит 2 — *MCR (Media Change Requested)* — индикатор запроса смены носителя. После обнаружения запроса смены носителя следующие команды *Door Lock* будут возвращать бит ошибки *ERR* и установленный бит *MCR*. Бит *MCR* сбрасывается командами *Door Unlock*, *Media Eject* или аппаратным сбросом.
- Бит 1 — *TK0NF (Track 0 Not Found)* — указывает на то, что по команде *Recalibrate* не удалось найти нулевой трек.
- Бит 0 — *AMNF (Address Mark Not Found)* — не найден адресный маркер данных в заголовке сектора.

После включения питания, сброса или команды *Execute Device Diagnostic* регистр ошибок содержит диагностический код.

Регистр свойств FR используется в зависимости от команды. В команде *Set Features* через него задается код подкоманды. Некоторые устройства могут игнорировать запись в этот регистр. До принятия ATA-2 в этот регистр помещали значение рекомендуемого номера цилиндра для предкомпенсации записи.

Регистр счетчика секторов SC содержит число секторов, участвующих в обмене. Хост инициализирует этот регистр до подачи команды (нулевое значение соответствует 256 секторам). По успешному завершению операции доступа к данным регистр должен обнулиться. Если команда завершилась с ошибкой, в регистре будет число секторов, которые должны быть переданы для успешного завершения предыдущего запроса. Команды *Initialize Device Parameters* или *Write Same* могут переопределить значение этого регистра.

6.3. Система команд

Стандарт ATA задает *систему команд*, ориентированную на накопители на магнитных дисках. В табл. 6.4 графа «Протокол» определяет протокол передачи данных, требуемых для команды (см. 6.5): *PI* — ввод данных от устройства в режиме PIO, *PO* — вывод данных в устройство в режиме PIO, *P* — протокол передачи командного пакета (16 байт вывода в режиме PIO), *DM* — обмен данными по каналу DMA, *DMQ* — обмен DMA с очередями, *ND* — нет обмена данными, *DR* — протокол сброса, *VS* — специфично для устройства. Графа «Тип» определяет характеристику команды: *O* — обязательная для всех устройств, *П* — обязательная для ATAPI, *Д* — дополнительная, *С* — специфическая. Графа «Код» содержит шестнадцатеричный код, загружаемый в регистр команды. Коды, помеченные звездочкой, поддерживаются для совместимости со старым ПО. В графе «Использование регистров» приняты следующие обозначения: *CY* — регистры цилиндров, *SC* — регистр счетчика секторов, *DH* — регистр номера устройства и головки, *SN* — регистр номера сектора, *FR* — регистр свойств. Назначение регистров: *у* — регистр содержит параметры для команды (для регистра номера устройства и головки используются оба пара-

метра), *D* — используется только номер устройства, *d* — задействован только номер устройства, использование поля номера головки специфично для производителя, *D0* — команда адресуется к ведущему устройству, но выполняют ее оба устройства.

Таблица 6.4. Система команд ATA

Команда	Протокол	Тип	Код	Использование регистров				
				FR	SC	SN	CY	DH
Acknowledge Media Change — подтверждение смены носителя[1]	VS	Д	DBh	–	–	–	–	D
Boot – Post-Boot — резерв для сменных носителей[1]	VS	Д	DCh	–	–	–	–	D
Boot – Pre-Boot — резерв для сменных носителей[1]	VS	Д	DDh	–	–	–	–	D
CFA Erase Sectors — стирание секторов[2]	ND	C	C0h	–	y	y	y	y
CFA Request Extended Error Code — чтение расширенного кода ошибки[2]	ND	C	03h	–	y	y	y	D
CFA Translate Sector — получение информации о состоянии сектора[2]	PI	C	87h	–	–	y	y	y
CFA Write Multiple Without Erase — множественная запись без операции стирания (в предварительно стертые секторы)[2]	PO	C	CDh	–	y	y	y	y
Check Power Mode — проверка режима энергопотребления	ND	Д	98h E5h	–	y	–	–	D
Door Lock — запретить смену носителя[3]	VS	Д	DEh	–	–	–	–	D

Таблица 6.4 (продолжение)

Команда	Протокол	Тип	Код	Использование регистров				
				FR	SC	SN	CY	DH
Door Unlock – разрешить смену носителя[3]	VS	Д	DFh	–	–	–	–	D
Device Reset – сброс устройства ATAPI[2]	DR	П	08h	–	–	–	–	D
Download Microcode – загрузка микрокода внутреннего ПО[4]	PO	Д	92h	y	y	y	y	D
Execute Device Diagnostic – диагностика	ND	О	90h	–	–	–	–	D08
Flush Cache – выгрузка кэша записи на носитель (время выполнения может достигать 30 с)[2]	ND	Д	E7h	–	–	–	–	D
Format Track – форматирование трека[3]	VS	С	50h	–	–	–	–	D8
Get Media Status – получение информации о состоянии носителя (наличие, смена, защита записи)[2]	ND	Д	DAh	–	–	–	–	D
Identify Device – идентификация устройства (не пакетного)	PI	О	Ech	–	–	–	–	D
Identify Device DMA – идентификация устройства с ответом в режиме DMA[5]	DM	Д	EEh	–	–	–	–	D
Identify Packet Device – идентификация устройства (пакетного)[2]	PI	П	A1h	–	–	–	–	D

Команда	Протокол	Тип	Код	Использование регистров				
				FR	SC	SN	CY	DH
Idle — переход в состояние ожидания	ND	Д	97h E3h	–	y	–	–	D
Idle Immediate — немедленный переход в состояние ожидания	ND	Д	95h E1h	–	–	–	–	D
Initialize Device Parameters — инициализация параметров устройства	ND	O	91h	–	y	–	–	y
Media Eject — извлечь носитель[4]	ND	Д	EDh	–	–	–	–	D
Media Lock — заблокировать носитель (не ATAPI)[2]	ND	Д	DEh	–	–	–	–	D
Media Unlock — разблокировать носитель (не ATAPI)[2]	ND	Д	DFh	–	–	–	–	D
Nop — холостая команда	ND	Д	00h	–	–	–	–	y
Packet — посылка командного пакета[2]	P	П	A0h	y	y	y	y	D
Read Buffer — чтение буфера	PI	Д	E4h	–	–	–	–	D
Read DMA (w/retry) — чтение по DMA с повторами	DM	Д	C8h	–	y	y	y	y
Read DMA (w/o retry) — чтение по DMA без повторов	DM	Д	C9h	–	y	y	y	y
Read DMA Queued — чтение по DMA с возможностью постановки команды в очередь и освобождения шины[2]	DMQ	Д	C7h	y	y[8]	y	y	y

Таблица 6.4 (продолжение)

Команда	Протокол	Тип	Код	Использование регистров				
				FR	SC	SN	CY	DH
Read Long (w/retry) — «длинное» чтение с повторами[3]	PI	Д	22h	–	у	у	у	у
Read Long (w/o retry) — «длинное» чтение без повторов[3]	PI	Д	23h	–	у	у	у	у
Read Multiple — множественное чтение	PI	Д	C4h	–	у	у	у	у
Read Native Address — чтение максимального адреса сектора (заводской установки)[2]	ND	Д	F8h	–	–	–	–	D
Read Sector(s) (w/retry) — чтение секторов с повторами	PI	О	20h	–	у	у	у	у
Read Sector(s) (w/o retry) — чтение секторов без повторов	PI	О	21h	–	у	у	у	у
Read Verify Sector(s) (w/retry) — верификационное чтение секторов с повторами	ND	О	40h	–	у	у	у	у
Read Verify Sector(s) (w/o retry) — верификационное чтение секторов без повторов	ND	О	41h	–	у	у	у	у
Recalibrate — рекалибровка (поиск нулевого трека)[3]	ND	Д	1xh	–	–	–	–	D
Security Disable Password — отмена защиты (с посылкой пароля)[6]	PO	Д	F6h	–	–	–	–	D

Команда	Протокол	Тип	Код	Использование регистров				
				FR	SC	SN	CY	DH
Security Erase Prepare — подготовка к защитному стиранию[6]	ND	Д	F3h	–	–	–	–	D
Security Erase Unit — защитное стирание (с посылкой пароля)[6]	PO	Д	F4h	–	–	–	–	D
Security Freeze Lock — блокировка команд защиты[6]	ND	Д	F5h	–	–	–	–	D
Security Set Password — установка пароля защиты[6]	PO	Д	F1h	–	–	–	–	D
Security Unlock — снятие защиты[6]	PO	Д	F1h	–	–	–	–	D
Seek — поиск	ND	О	7xh	–	–	у	у	у
Service — передача данных и/или состояния команды, освободившей шину[2]	P или DMQ	Д	A2h	–	–	–	–	D
Set Features — установка свойств	ND	Д	Efh	у	–	–	–	D
Set Max Address — установка максимального адреса[2]	ND	Д	F9h	–	у[8]	у	у	у
Set Multiple Mode — установка множественного режима	ND	Д	C6h	–	у	–	–	D
Sleep — перевод в «спящий» режим	ND	Д	99h E6h	–	–	–	–	D
SMART — группа команд мониторинга[6]	ND, PI	Д	B0h	у	у[8]	у[8]	у[8]	D
Standby — перевод в дежурный режим	ND	Д	96h E2h	–	у	–	–	D
Standby Immediate — немедленный перевод в дежурный режим	ND	Д	94h E0h	–	–	–	–	D

Таблица 6.4 (*продолжение*)

Команда	Протокол	Тип	Код	Использование регистров				
				FR	SC	SN	CY	DH
Write Buffer — запись в буфер	PO	Д	E8h	–	–	–	–	D
Write DMA (w/retry) — запись по DMA с повторами	DM	Д	CAh	–	у	у	у	у
Write DMA (w/o retry) — запись по DMA без повторов	DM	Д	CBh	–	у	у	у	у
Write DMA Queued — запись по DMA с возможностью постановки команды в очередь и освобождения шины[2]	DMQ	Д	CCh	у	у[8]	у	у	у
Write Long (w/retry) — «длинная» запись с повторами[3]	PO	Д	32h[7]	–	у	у	у	у
Write Long (w/o retry) «длинная» запись без повторов[3]	PO	Д	33h[7]	–	у	у	у	у
Write Multiple — множественная запись	PO	Д	C5h[7]	–	у	у	у	у
Write Same — запись с размножением	PO	Д	E9h	у	у	у	у	у
Write Sector(s) (with retry) — запись секторов с повторами	PO	O	30h[7]	–	у	у	у	у
Write Sector(s) (without retry) — запись секторов без повторов	PO	O	31h[7]	–	у	у	у	у
Write Verify — запись с верификацией	PO	Д	3Ch[7]	–	у	у	у	у

[1] В ATA-3+ команда не определена.

[2] Только для ATA-4.

[3] В АТА-4 команда не определена.

[4] Для АТА-2 и выше.

[5] Только для АТА-3.

[6] Для АТА-3 и выше.

[7] Коды поддерживаются для совместимости.

[8] Использование регистра зависит от производителя.

6.3.1. Основные команды

К этой группе относятся команды АТА для доступа к данным. Для устройств, существенно отличающихся от НЖМД, имеется альтернативный способ доставки команд и параметров — команда *Packet*. По этой команде, используя вывод в режиме PIO, хост-адаптер передает ***пакет с блоком дескриптора команды***. Подробнее о пакетном интерфейсе см. в 6.7.

Минимальной адресуемой единицей ***команд чтения и записи*** является 512-байтный сектор.

Команды чтения секторов в режимах обмена PIO — *Read Sector(s)* и DMA — *Read DMA* позволяют считывать последовательно расположенные секторы, количество которых задано в регистре *SC*, а адрес начального сектора — в регистрах *CH*, *CL*, *DH* и *SN*. Команды чтения бывают с повторами или без них. В первом случае, если при чтении сектора обнаружена неисправимая ошибка, устройство автоматически делает несколько повторных попыток чтения. После исполнения команды в случае неисправимой ошибки блок командных регистров содержит адрес сектора с ошибкой. Если устройство использует ЕСС-код, то некоторые ошибки чтения оно исправляет без повторов, но указывает на этот факт в регистре ошибок.

Команда чтения в блочном режиме передачи *Read Multiple* отличается от обычного (с обменом PIO) тем, что запросы прерывания вырабатываются не на каждый сектор, а на блок секторов, размер которого задан командой *Set Multiple Mode*. Кроме того, в случае появления неисправимой ошибки чтения содержимое блока командных регистров не определено (не указывает на сбойный сектор). За счет сокращения числа прерываний, которые должен обслужить процессор, блочный режим в многозадачной системе позволяет повысить

производительность на 30%. Производительность обмена зависит от размера блока. Размер, оптимальный для устройства, может не быть оптимальным для ОС.

Команда «длинного» чтения *Read Long* считывает сектор данных вместе с контрольными байтами. При ее вызове регистр *SC* должен указывать на запрос только одного сектора. Блок данных считывается 16-битными словами, а контрольные байты — 8-битными. Некоторые устройства АТА-1 неспособны быстро передавать байты ЕСС вслед за данными. Для их считывания используют 8-битный обмен в режиме PIO Mode 0.

Команда верификации *Read Verify Sector(s)* в отличие от обычного чтения не передает данные от устройства. В случае ошибки на адрес сбойного сектора указывает блок командных регистров. Запрос прерывания выполняется после исполнения команды.

Команды записи *Write DMA*, *Write Sector(s)*, *Write Long*, *Write Multiple* работают аналогично соответствующим им командам чтения и также имеют версии с повторами и без. Команда записи с верификацией *Write Verify* аналогична команде *Write Sector(s)*, но для каждого сектора после записи выполняется контрольное считывание. Для логической инициализации (очистки области) дисков имеется команда записи *Write Same*, которая позволяет содержимое 512 байт, принятых от хоста, записать в группу секторов. Если в регистр свойств занесен код 22h, область записываемых секторов определяется регистрами командного блока. Если в регистр свойств занесен код DDh, запись производится во все доступные секторы устройства.

Команды чтения и записи буфера *Read Buffer* и *Write Buffer* служат для обмена в режиме PIO с 512-байтной буферной памятью устройства (но не сектором носителя).

Команда форматирования трека *Format Track* по входным параметрам специфична для каждого устройства, и ее использование не рекомендуется. Многие устройства ее отвергают как недопустимую. Поскольку команда форматирования традиционно ориентирована на один трек, форматирование дисков АТА в режиме трансляции геометрии, при котором физическая

организация не совпадает с логической, невозможно. В накопителях с зонным форматом записи форматирование логического трека будет «накрывать» переменное число физических.

Вспомогательное назначение имеют *команды поиска*. По команде поиска *Seek* устройство устанавливает головки на заданный цилиндр/трек и считывает идентификатор сектора.

Команда рекалибровки *Recalibrate* заставляет устройство найти нулевой цилиндр. Если устройству это не удается, устанавливается бит ошибки *Track 0 Not Found*. Эту команду обычно применяют при обработке ошибок: часто после такого «встряхивания» ошибка не повторяется. После успешного выполнения команды блок командных регистров содержит адрес первого сектора диска в формате, зависящем от режима адресации (LBA или CHS).

Для *накопителей со сменными носителями* в ATA-2 были предназначены команды загрузки и выгрузки, подтверждения смены носителя, блокировки и разблокировки устройства — *Boot — Post-Boot*, *Boot — Pre-Boot*, *Media Eject*, *Acknowledge Media Change*, *Door Lock*, *Door Unlock*. Их реализация специфична для каждой модели устройства. В ATA-4 набор этих команд сокращен. Смену носителя вызывает только команда *Media Eject* (для устройств ATAPI — пакетная команда *Start/Stop Unit*). Здесь может работать один из двух механизмов защиты от несанкционированной смены носителя. При использовании уведомления о смене носителя *Removable Media Status Notification* нажатие кнопки на накопителе смены носителя не вызывает, этот факт и состояние носителя могут определяться по команде *Get Media Status*. Этот механизм включается и отключается соответствующими подкомандами *Set Features*; по любому сбросу устройства он отключается. Когда механизм уведомления отключен, работает другой: для запрета/разрешения смены носителей от кнопки устройства (только для непакетных устройств) применяют команды *Media Lock/Media Unlock*, они же используются для определения состояния (наличия, защиты записи и факта смены носителя).

Для *запоминающих устройств на флэш-памяти* в ATA-4 ввели команды, начинающиеся с аббревиатуры CFA (Compact

Flash Association — ассоциация производителей компактных флэш-карт). Специфика этих устройств заключается в записи: запись обеспечивается лишь в предварительно стертые ячейки (сектор), хотя есть устройства, автоматически осуществляющие стирание при записи. Операция записи выполняется существенно медленнее, чем чтения, скорость которого приближается к скорости динамической памяти. Операция стирания занимает еще больше времени. Интерес представляет информация о состоянии сектора: стертый или нет и сколько раз выполнялась запись.

6.3.2. Служебные команды

В системе команд имеются средства *идентификации и управления свойствами* устройств. Команда идентификации *Identify Device* позволяет считать из контроллера блок из 256 слов, характеризующих устройство ATA. Устройства с пакетным интерфейсом эту команду должны отвергать, для них имеется команда *Identify Packet Device* (см. 6.7). Блок параметров может храниться как в энергонезависимой памяти устройства, так и на самом носителе в месте, недоступном для обычных обращений. Состав полей блока для непакетных устройств приведен в табл. 6.5. Графа «*F/V*» определяет свойства соответствующих полей: *F* — содержимое фиксировано, *V* — содержимое меняется в зависимости от состояния устройства или выполненных команд, *X* — специфично для данного устройства (может быть и *F*, и *V*), *R* — зарезервированные поля. Для устройств со сменными носителями значения полей *F* могут меняться при смене носителя.

Таблица 6.5. Информация идентификации устройства ATA

Слово	F/V	Назначение
0	F	*Общая конфигурационная информация* Бит 15: 0 для всех устройств ATA Бит 7: 1 — устройство со сменным носителем Бит 6: 1 — устройство (контроллер) с несменным носителем Бит 0 — зарезервирован Значение остальных бит специфично для устройств
1	F	*Число логических цилиндров*

Слово	F/V	Назначение
2	R	Зарезервировано
3	F	*Число логических головок*
4	X	Специфично
5	X	Специфично
6	F	*Число логических секторов на логическом треке*
7–9	X	Специфично
10–19	F	*Серийный номер* (20 символов ASCII), если слово 10 ненулевое, иначе — специфичная информация
20	X	Специфично
21	X	Специфично
22	F	*Число дополнительных байт* в командах «длинного» чтения и записи (в ATA-4 объявлено устаревшим)
23–26	F	*Версия встроенного ПО* (8 символов ASCII), если слово 23 ненулевое, иначе — специфичная информация
27–46	F	*Номер модели* (40 символов ASCII), если слово 27 ненулевое, иначе — специфичная информация
47	X	Биты 15–8 специфичны (в ATA-4 — 80h)
	F	Биты 7–0 — возможности множественных операций: 0 — зарезервирован, 01h–FFh — максимальное количество секторов, передаваемых за сеанс
48	R	Зарезервировано
49		*Возможности:*
	R	Биты 15–14 — зарезервированы для *Identify Packet Device*
	F	Бит 13 — значения таймера режима Standby: 1 — соответствуют стандарту, 0 — задаются особо
	R	Бит 12 зарезервирован для *Identify Packet Device*
	F	Бит 11: 1 — IORDY поддерживается (для PIO Mode 3 и выше — обязательно), 0 — IORDY может поддерживаться
	F	Бит 10: 1 — сигнал IORDY может быть запрещен командой *Set Features*
	F	Бит 9 =1 (в ATA-4 используется для *Identify Packet Device*, в ATA-2 — как указание на поддержку LBA и действительности слов 60–61)

Таблица 6.5 (продолжение)

Слово	F/V	Назначение
	F	Бит 8=1 (в АТА-4 используется для *Identify Packet Device*, в АТА-2 — как указание на поддержку DMA)
	X	Биты 7–0 специфичны
50	F	*Возможности* (АТА-4): Бит 15=0 Бит 14=1 Биты 13–1 — зарезервирован Бит 0=1 указывает на специфичное значение минимума для Standby Timer
51	F	Биты 15–8: *длительность цикла обмена в режиме PIO*. Если значение параметра не соответствует режимам 0, 1 или 2, используется PIO Mode 0
	X	Биты 7–0 специфичны
52	F	Специфично. В АТА-2 биты 15–8: временной режим цикла одиночного обмена DMA (Single Word DMA). Если значение параметра не соответствует режимам 0, 1 или 2, используется Mode 0. Если поддерживаются слова 62 или 63, значение поля игнорируется
	X	Биты 7–0 специфичны
53	R	Биты 15–3 зарезервированы
	F	Бит 2: 1 — поля в слове 88 действительны, 0 — нет
	F	Бит 1: 1 — поля в словах 64–70 действительны, 0 — нет. Любое устройство, поддерживающее PIO Mode 3 и старше или Multiword DMA Mode 1 и выше, должно использовать эти поля
	V	Бит 0: 1 — поля в словах 54–58 действительны, 0 — поля могут быть действительны
54	V	*Текущее число логических цилиндров*. Для устройств АТА-1, если команда *Initialize Device Parameters* не применялась, значение слова специфично
55	V	*Текущее число логических головок*. Для устройств АТА-1, если команда *Initialize Device Parameters* не применялась, значение слова специфично

Слово	F/V	Назначение
56	V	*Текущее число логических секторов на трек.* Для устройств ATA-1, если команда *Initialize Device Parameters* не применялась, значение слова специфично
57–58	V	*Текущая емкость в секторах* (произведение слов 54, 55, 56)
59	R	Биты 15–9 зарезервированы
	V	Бит 8: 1 — установка для многосекторной передачи действительна
	V	Биты 7–0: текущая установка максимального числа секторов для многосекторных команд
60–61	F	*Общее число секторов*, адресуемых пользователем (только в режиме LBA)
62		Для ATA-4 специфично, в ATA-2 — *режим одиночного обмена DMA*:
	V	Биты 15–8: активный режим: бит 8=1 — Mode 0, бит 9=1 — Mode 1 и т. д. Единичное значение может иметь только один бит
	F	Биты 7–0 — поддерживаемые режимы: бит 0=1 — Mode 0, бит 1=1 — Mode 1 и т. д.
63		*Режим множественного обмена Multiword DMA*:
	V	Биты 15–8: активный режим, аналогично слову 62
	F	Биты 7–0: поддерживаемые режимы, аналогично слову 62
64	R	Биты 15–8 зарезервированы
	F	Биты 7–0 — *поддерживаемые прогрессивные режимы программного обмена (Advanced PIO)*: бит 0=1 — PIO Mode 3, бит 1=1 — PIO Mode 4, биты 2–7 — зарезервированы
65	F	*Минимальная длительность цикла передачи множественного DMA* (в наносекундах)
66	F	*Рекомендованная длительность цикла передачи множественного DMA* (в наносекундах). При многосекторной передаче обеспечивает оптимальную скорость, при которой устройство не будет тормозить поток снятием запроса DMARQ

Таблица 6.5 (*продолжение*)

Слово	F/V	Назначение
67	F	*Минимальная длительность цикла передачи PIO без использования сигнала готовности* (в наносекундах)
68	F	*Минимальная длительность цикла передачи PIO с использованием сигнала готовности* (в наносекундах)
69–70	R	Зарезервировано для поддержки перекрытия команд и очередей
71–74	R	Зарезервировано для команды *Identify Packet Device*
75	F	*Глубина очереди команд*: Биты 15–5 — зарезервированы Биты 4–0 — максимальная глубина очереди
76–79	R	Зарезервировано
80	F	*Номер основной версии* интерфейса (если не 0000h или FFFFh): 1 – ATA-1, 2 – ATA-2, 4 – ATA/ATAPI-4...
81	F	*Младший номер версии* интерфейса (если не 0000h или FFFFh)
82	F	*Поддержка команд и свойств* (если слова 82 и 83 не равны 0000h или FFFFh): Бит 15 — не используется Бит 14 — команды *Nop* Бит 13 — команды *Read Buffer* Бит 12 — команды *Write Buffer* Бит 11 — не используется Бит 10 — ограничение доступного пространства Бит 9 — команда *Device Reset* Бит 8 — вырабатывание прерывания во время выполнения команды *Service* Бит 7 — прерывание по освобождению шины Бит 6 — кэширование с упреждающим чтением Бит 5 — кэширование записи Бит 4 — команда *Packet* Бит 3 — управление энергопотреблением Бит 2 — смена носителя Бит 1 — команды *Security* Бит 0 — *SMART*

Слово	F/V	Назначение
83	F	*Поддержка команд и свойств* (как и слово 82): Бит 15 = 0 Бит 14 = 1 Биты 13–5 зарезервированы Бит 4 — уведомление о смене носителя Бит 3 — расширенное управление энергопотреблением Бит 2 — команды *CFA* Бит 1 — команды *Read/Write DMA Queued* Бит 0 — команда *Download Microcode*
84	F	*Поддержка команд и свойств* (если слова 82, 83 и 84 не равны 0000h или FFFFh): Бит 15 = 0 Бит 14 = 1 Биты 13–0 зарезервированы
85–87	F	*Разрешенные команды и свойства*; поля и правила аналогичны словам 82–84
88	R R V	*Режим Ultra DMA* Биты 15–11 зарезервированы Биты 10–8: активный режим Ultra DMA: бит 8=1 — Mode 0, бит 9=1 — Mode 1 и т. д. Единичное значение может иметь только один бит Биты 7–3 зарезервированы Биты 2–0: поддерживаемые режимы одиночного обмена DMA: бит 0=1 — Mode 0, бит 1=1 — Mode 1 и т. д.
89	F	*Время защитного стирания*: 0 — не указано; значение 1 — 254, умноженное на 2, дает время в минутах, 255 — более 508 мин
90	F	*Время расширенного защитного стирания* (аналогично слову 89)
91	V	*Текущий уровень Advanced Power Management*
92–126	R	Зарезервировано
127	R F	Биты 15–2 — зарезервированы Биты 1–0 — *уведомление о смене носителя*: 00 — не поддерживается, 01 — поддерживается, 1x — зарезервированы

Таблица 6.5 (продолжение)

Слово	F/V	Назначение
128	V	Состояние защиты: Биты 15–9 зарезервированы Бит 8 — уровень защиты: 0 — высокий, 1 — максимальный Биты 7–6 — зарезервированы Бит 5: 1 — поддержка расширенного защитного стирания Бит 4: 1 — счетчик попыток иссяк Бит 3: 1 — команды защиты блокированы Бит 2: 1 — устройство заблокировано Бит 1: 1 — защита разрешена Бит 0: 1 — защита поддерживается
129–159	X	Специфично
160–255	R	Зарезервировано

Команда установки параметров *Initialize Device Parameters* задает режим трансляции логической геометрии в системе CHS. Регистр *SC* указывает число логических секторов на трек, а в поле номера головки регистра *DH* указывается уменьшенное на единицу число логических головок. Если запрошенный режим трансляции устройство не поддерживает, оно выдаст ошибку «команда отвергнута» (прежняя спецификация ATA не четко описывала эту ситуацию, и некоторые системы не обнаруживали отказ при данной команде). После запроса неподдерживаемого режима трансляции устройство блокирует доступ к носителю с ошибкой *ID Not Found* до запроса поддерживаемого режима трансляции. Устройство обязано поддерживать режим, описанный словами 1, 3 и 6 блока его параметров. Некоторые устройства ATA-1 требовали, чтобы данная команда выполнялась до доступа к носителю.

Команда установки свойств *Set Features* (специфичная для устройства) подразумевает помещение кода подкоманды (табл. 6.6) в регистр свойств.

Таблица 6.6. Подкоманды задания свойств устройства

Код	Назначение
01h[1]	Разрешение 8-битного обмена
02h	Разрешение кэширования записи

Код	**Назначение**
03h	Установка режима передачи по значению регистра *SC* (табл. 6.7)
04h[1]	Разрешение автоматического переназначения дефектных блоков
05h	Разрешение расширенного управления энергопотреблением (уровень в регистре *SC*)
31h	Запрет уведомления о смене носителя
33h[1]	Запрет повторов
44h[1]	Установка количества дополнительных байт в командах *Read Long/Write Long*
54h[1]	Установка количества сегментов кэша по значению регистра *SC*
55h	Запрет упреждающего чтения
5Dh	Разрешение прерывания по освобождению шины
5Eh	Разрешение прерывания при выполнении команды *Service*
66h	Запрет возврата к параметрам по умолчанию при включении питания
77h[1]	Запрет ECC
81h[1]	Запрет 8-битного обмена
82h	Запрет кэширования записи
84h[1]	Запрет автоматического переназначения дефектных блоков
85h	Запрет расширенного управления энергопотреблением
88h[1]	Разрешение ECC
99h[1]	Разрешение повторов
9Ah[1]	Ограничение тока потребления значением в регистре *SC*, умноженным на 4 (мА)
AAh	Разрешение упреждающего чтения
ABh[1]	Установка максимального количества кэшируемых блоков по регистру *SC*
BBh[1]	Установка 4-байтной длины дополнительного поля для команд *Read Long/Write Long*
CCh	Разрешение возврата к параметрам по умолчанию при включении питания
DDh	Запрет прерывания по освобождению шины
DEh	Запрет прерывания при выполнении команды *Service*

[1] В ATA-4 подкоманда изъята.

После включения питания или аппаратного сброса установленные свойства заменяются на принятые по умолчанию. Результат подкоманд можно и закрепить, подав подкоманду 66h (отмена — подкоманда с кодом CCh).

Таблица 6.7. Управление режимом обмена

Биты [7:0]	Режим обмена
00000 000	Режим PIO, принятый для устройства по умолчанию
00000 001	Режим PIO, принятый для устройства по умолчанию; запрет IORDY
00001 *nnn*	Режим PIO Mode x* с сигналом IORDY
00010 *nnn*	Режим одиночного DMA Mode x
00100 *nnn*	Режим множественного DMA Mode x
01000 *nnn*	Режим Ultra DMA Mode x
10000 *nnn*	Зарезервированы

* x определяется значением поля *nnn*

Команда задания параметров блочного режима передачи *Set Multiple Mode* через регистр *SC* указывает число секторов, передаваемых с одним запросом прерывания. Значению *SC*=0 соответствует запрет блочного режима. После включения питания или аппаратного сброса блочный режим запрещен.

Команда диагностики Execute Device Diagnostic, адресуясь всегда к ведущему устройству, выполняется одновременно обоими устройствами. О ее результате ведомое устройство сообщает ведущему (сигналом PDIAG#). Состояние обоих устройств определяется по диагностическому коду (табл. 6.8), который считывается из бит [6:0] регистра ошибок нулевого устройства.

Фиктивная команда Nop, не изменяя содержимого регистров, позволяет считать информацию о состоянии устройства, выполнив всего одну 16-битную запись по адресу в регистре *DH*. Команда всегда возвращает бит ошибки «команда отвергнута». В АТА-4 команда объявлена обязательной для устройств ATAPI и всех, поддерживающих перекрывающиеся команды. Она имеет подкоманды, указываемые в регистре свойств. Подкоманда с кодом 00 сбрасывает всю оставшуюся очередь, а с кодом 01 (*NOP Auto Poll*) на очередь не влияет.

Таблица 6.8. Диагностические коды

Код	Ведущее устройство	Ведомое устройство
01h	Нормально	Нормально или отсутствует
00h, 02h-7Fh	Неисправно	Нормально или отсутствует
81h	Нормально	Неисправно
80h, 82h-FFh	Неисправно	Неисправно

6.3.3. Дополнительные сервисные функции

Устройства АТА и ATAPI могут поддерживать управление энергопотреблением, защищать данные от несанкционированного доступа, выполнять мониторинг своего состояния для предупреждения об угрозе отказа и перезаписывать встроенную микропрограмму.

Средства *управления энергопотреблением — Power Management* — не обязательны. Различают следующие состояния, перечисленные в порядке возрастания энергопотребления:

Sleep — «заснувшее» устройство — потребляет минимум энергии, «разбудить» его может только сброс. Время «пробуждения» — не более 30 с.

Standby Mode (дежурный режим) — устройство способно принимать команду по интерфейсу, но для доступа к носителю может потребоваться столь же большое время. В это состояние устройство может перейти как по команде, так и по таймеру (*Standby Timer*), отсчитывающему время от последнего запроса, полученного в состоянии ожидания или активном состоянии. Время срабатывания таймера программируется; он может быть запрещен.

Idle Mode (состояние ожидания) — устройство способно сразу начать обслуживание обращения к носителю, не слишком быстро, так как некоторые узлы отключены.

Active Mode (активный режим) — устройство все запросы обслуживает за кратчайшее время.

Команда проверки режима энергопотребления *Check Power Mode* через регистр *SC* возвращает состояние: *SC*=0 — устройство находится в состоянии *Standby* или переходит в него, *SC*=255 — устройство активно или в состоянии ожидания.

Команда *Idle* переводит устройство в режим ожидания из активного или дежурного режимов, а также через регистр *SC* программирует таймер дежурного режима (табл. 6.9).

Команда *Idle Immediate* непосредственного перевода в режим ожидания таймер не программирует.

Таблица 6.9. Тайм-аут перехода в дежурный режим

Содержимое SC	Тайм-аут
0 (00h)	Тайм-аут запрещен
x=1– 240 (01h–F0h)	(x×5) с
x=241– 251 (F1h–FBh)	((x–240)×30) мин
252 (FCh)	21 мин
253 (FDh)	Период, заданный производителем (в пределах 8–12 ч)
254 (FEh)	Зарезервировано
255 (FFh)	21 мин 15 с

Команда *Sleep* является единственным способом перевода устройства в «спящий» режим. После получения команды устройство генерирует запрос прерывания. Хост должен считать регистр состояния, что сбросит прерывание и позволит устройству «заснуть». «Разбудить» устройство можно только аппаратным или программным сбросом. Состояние, в которое оно перейдет, определяется типом сброса и программируется. Поскольку не каждый хост «знает», что после команды нужно прочитать регистр состояния, устройство автоматически сбросит запрос прерывания и «заснет» через определенный период времени (не менее 2 с).

Команда *Standby* переводит устройство в дежурный режим потребления и через параметр в регистре *SC* программирует таймер. Команда *Standby Immediate* таймер не программирует.

Устройства могут поддерживать расширенное управление энергопотреблением *APM (Advanced Power Management)*. При этом задается уровень *APM Level*, определяющий степень активности: 01h — минимальное потребление, FEh — максимальная производительность. Уровень выше 80h не позво-

ляет устройству останавливать двигатель. APM управляется подкомандами *Set Features*. Устройства с пакетным интерфейсом для управления энергопотреблением могут использовать также команды пакетного протокола.

Начиная с ATA-3 в стандарт введена *группа команд защиты*. Защищенное устройство по включению питания или аппаратному сбросу будет находиться в заблокированном состоянии, при котором любой доступ к информации на носителе запрещен. Система защиты поддерживает два пароля — *главный* (*Master Password*) и *пользовательский* (*User Password*). Главный пароль устанавливается изготовителем или продавцом устройства. Пароль устанавливается командой *Security Set Password*, этой же командой устанавливается защита. В передаваемом блоке данных (512 байт) нулевое слово является управляющим:

- Бит 0 определяет тип пароля (0 — User, 1 — Master);
- Бит 8 определяет степень защиты (0 — High, 1 — Maximum).

Слова 1–16 содержат пароль, остальные игнорируются. При установке главного пароля состояние защиты не меняется.

Разблокировать устройство позволяет команда *Security Unlock*, при этом необходимо указать пароль пользователя. Если пароль утерян, можно использовать главный пароль, но доступ к данным будет предоставлен, только если была выбрана степень защиты High. Если была выбрана степень защиты Maximum, разблокировать устройство по главному паролю можно только командой *Security Erase Unit*, но при этом вся информация с носителя будет стерта. Чтобы застраховаться от случайного стирания, непосредственно перед этой командой должна быть выполнена команда *Security Erase Prepare*. Команда *Security Freeze Lock* блокирует выполнение любых команд защиты до следующего сброса. Для осложнения подбора пароля (его длина составляет 32 байт) служит счетчик неудачных попыток разблокировки, по срабатывании которого команды разблокировки будут отвергаться до выключения питания или аппаратного сброса. Отменить защиту разблокированного устройства можно командой *Security Disable Password* (предъявив один из двух паролей).

Команды защиты в ATA-4 дополнены ограничением максимального адреса, доступного пользователю (сообщаемого в блоке параметров идентификации), командой *Set Max Address*. Узнать реальный максимальный адрес позволяет команда *Read Native Address*.

Для предупреждения о возможном отказе устройства служит технология *SMART* (Self-Monitoring, Analysis and Reporting Technology — технология самонаблюдения, анализа и сообщений). *Предсказуемые отказы* (Predictable Failure) появляются в результате выхода параметров за некоторый порог. Отслеживаемые параметры: время разгона до заданной скорости, время позиционирования, процент ошибок позиционирования, высота полета головок, производительность (зависящая от числа повторов), количество использованных резервных секторов и др. Мониторинг может осуществляться двояко: в рабочем режиме (*On-Line*) одновременно с выполнением команд хоста (при некотором возможном замедлении). Мониторинг *Off-Line* выполняется устройством в паузе между «полезными» командами, не снижая производительности. Если во время выполнения этой процедуры придет внешняя команда, то мониторинг прервется на время исполнения команды, но начало исполнения команды может задержаться на время до двух секунд. Значения атрибутов, за которыми ведется наблюдение, сохраняются в энергонезависимой памяти устройства.

Для непакетных устройств имеется команда *Smart* (пакетные используют для этих целей собственный протокол), подкоманды которой задаются через регистр свойств. Из этих подкоманд стандартизованы следующие:

- *Smart Read Data* (D0h) — чтение блока данных SMART;
- *Smart Enable/Disable Attribute Autosave* (D2h) — управление автосохранением атрибутов;
- *Smart Save Attribute Values* (D3h) — сохранение значений атрибутов;
- *Smart Execute Off-Line Immediate* (D4h) — немедленное выполнение мониторинга Off-Line;
- *Smart Enable Operations* (D8h) — разрешение команд и внутренних функций SMART;

- *Smart Disable Operations* (D9h) — запрет команд и функций SMART;
- *Smart Return Status* (DAh) — опрос результатов мониторинга. Если какой-либо порог перейден, регистры *CH* и *CL* будут иметь значения C2h и 4Fh соответственно, если все в порядке — 2Ch и F4h.

Команда загрузки микрокода Download Microcode позволяет модифицировать *firmware* — встроенное ПО устройства. В зависимости от кода в регистре свойств загруженный микрокод будет действовать до выключения питания (*FR*=01h) или постоянно (*FR*=07h). Количество загруженных блоков задается регистром *SN* (старший байт) и *SC* (младший байт). Таким образом может быть загружено от 0 до 33 553 920 байт, что более чем достаточно для встроенного ПО. Некорректная модификация кода может привести к выходу устройства из строя.

6.4. Протоколы взаимодействия хоста и устройства

Обычный протокол взаимодействия хоста с устройством выглядит следующим образом:

1. Хост читает регистр состояния устройства, дожидаясь нулевого значения бита *BSY*. Если присутствуют два устройства, хост обращается к ним «наугад» — состояние будет сообщать последнее выбранное устройство.
2. Дождавшись освобождения устройства, хост записывает в регистр *DH* байт, у которого бит *DEV* указывает на адресуемое устройство. Здесь кроется причина невозможности параллельной работы двух устройств на одной шине ATA: обратиться к устройству можно только после освобождения обоих устройств.
3. Хост читает основной или альтернативный регистр состояния адресованного устройства, дожидаясь признака готовности (*DRDY*=1).
4. Хост заносит требуемые параметры в блок командных регистров.

5. Хост записывает код команды в регистр команд.
6. Устройство устанавливает бит *BSY* и переходит к исполнению команды. Дальнейшие действия зависят от протокола передачи данных, заданного командой (см. графу «Протокол» в табл. 6.4).

Для команд, *не требующих передачи данных* (*ND*):

7. Завершив исполнение команды, устройство сбрасывает бит *BSY* и устанавливает запрос прерывания (если он не запрещен). К этому моменту в регистрах состояния и ошибок уже имеется информация о результате исполнения. Выполнение завершается.

Единичное значение бита *BSY* может промелькнуть между шагами 6 и 7 так быстро, что хост его не зафиксирует, но для фиксации факта выполнения команды или ее части и предназначен запрос прерывания.

Для команд, требующих *чтения данных в режиме PIO* (*PI*):

7. Подготовившись к передаче первого блока данных по шине АТА, устройство устанавливает бит *DRQ*. Если была ошибка, она фиксируется в регистрах состояния и ошибок. Далее устройство сбрасывает бит *BSY* и устанавливает запрос прерывания (если он не запрещен).
8. Зафиксировав обнуление бита *BSY* (или по прерыванию), хост считывает регистр состояния, что приводит к сбросу прерывания от устройства.
9. Если хост обнаружил единичное значение бита *DRQ*, он производит чтение первого блока данных в режиме PIO (адресуясь к регистру данных). Если обнаружена ошибка, считанные данные могут быть недостоверными.

После передачи блока данных возможно одно из следующих действий:

- Если на шаге 8 ошибка не обнаружена, а требуется передача следующего блока, устройство устанавливает бит *BSY*, и данная последовательность повторяется с шага 7.

- Если есть ошибка или передан последний блок данных, устройство сбрасывает бит *DRQ* и выполнение команды завершается.

Для *операций записи данных* после шага 6 для устройства начинается активная фаза записи на носитель, что отмечается установкой бита *BSY*.

Для команд, требующих *записи данных в режиме PIO* (*PO* и *P*):

7. Подготовившись к приему первого блока данных по шине ATA, устройство устанавливает бит *DRQ* (если нет ошибок) и сбрасывает бит *BSY*. Если была ошибка, она фиксируется.
8. Зафиксировав обнуление бита *BSY*, хост считывает регистр состояния.
9. Если хост обнаружил единичное значение бита *DRQ*, он производит запись первого блока данных в режиме PIO по адресу в регистре данных.
10. После передачи блока данных возможно одно из следующих действий:

- Если обнаружена ошибка, устройство сбрасывает бит *DRQ*, устанавливает запрос прерывания и выполнение команды завершается. Переданные по шине данные остаются необработанными устройством (не записываются на носитель).
- Если ошибка не обнаружена, устройство устанавливает бит *BSY* и переходит к следующему шагу.

11. Устройство обрабатывает принятый блок данных, затем:

- если нет ошибок и обработанный блок — последний, устройство сбрасывает бит *BSY* и устанавливает запрос прерывания, на чем выполнение команды успешно завершается;
- если обнаружена ошибка, выполнение команды завершается таким же образом, но с установкой бит ошибок;
- если нет ошибок и требуется передача следующего блока, выполняются следующие шаги:

12. По готовности приема следующего блока устройство устанавливает бит *DRQ*, сбрасывает бит *BSY* и устанавливает запрос прерывания.

13. По обнулению бита *BSY* (или по прерыванию) хост считывает регистр состояния.

14. Обнаружив бит *DRQ*, хост выполняет запись очередного блока в регистр данных, и последовательность повторяется с шага 11.

Команды с *передачей данных в режиме DMA* выполняются похожим образом, но:

- Вместо PIO используется прямой доступ к памяти. Хост должен проинициализировать канал DMA до записи кода в регистр команд, чтобы по появлении сигнала *DMARQ* начался обмен.
- Запрос прерывания даже в многосекторных передачах производится один раз — по выполнении команды.

6.5. Протоколы и режимы передачи данных

Программа общается с устройствами ATA через регистры, используя инструкции ввода/вывода *IN* и *OUT*. Для передачи данных с максимальной скоростью применяют программный доступ к регистру данных или DMA. Тип обмена задается командой. Программный доступ обязателен для всех устройств. Команды режима DMA устройствами могут не поддерживаться.

Программный доступ PIO (Programmed Input/Output) выполняется в виде следующих друг за другом операций чтения или записи в пространстве ввода/вывода по адресу регистра данных. В отличие от программно-управляемого ввода/вывода, применяемого, например, для общения с LPT-портом, передача блока данных в режиме PIO производится без программного опроса какого-либо бита готовности для передачи каждого слова. Готовность устройства проверяется перед началом передачи блока, после чего хост производит серию операций в определенном темпе. Темп определяется выбранным режимом *PIO Mode*. Для режимов определены

допустимые параметры временной диаграммы цикла обмена (табл. 6.10). Обмен PIO программно реализуется с помощью инструкций ввода/вывода строк *REP INS* или *REP OUTS* с занесенным в регистр *CX* количеством слов (или байт) в передаваемом блоке. Эти инструкции обеспечивают максимально возможную скорость обмена для данного процессора и системной шины. «Обуздать» процессор в соответствии с выбранным режимом входит в задачу адаптера ATA, который использует для удлинения цикла сигнал готовности шины (для ISA — IOCHRDY). Традиционные режимы 0, 1 и 2 имеют временные параметры, фиксируемые только хост-адаптером. Для прогрессивных режимов ATA-2 (PIO Mode 3 и старше) устройство может затормозить обмен, используя сигнал готовности IORDY. Программный обмен на все время передачи блока занимает и процессор, и системную шину.

Таблица 6.10. Параметры режимов передачи (PIO mode)

PIO mode	Минимальное время цикла, нс	Скорость передачи, Мбайт/с	Интерфейс
0	600	3,3	ATA
1	383	5,2	ATA
2	240	8,3	ATA
3	180	11,1	E-IDE, ATA-2 (используется IORDY)
4	120	16,6	E-IDE, Fast ATA-2 (используется IORDY)

Обмен по каналу DMA занимает исключительно шины ввода/вывода и памяти. Процессору требуется выполнить только процедуру инициализации канала, после чего до прерывания от устройства в конце передачи блока он свободен (этим могут воспользоваться многозадачные системы). Стандартные каналы DMA шины ISA для интерфейса ATA практически не используются из-за низкой пропускной способности. Высокопроизводительные адаптеры ATA могут иметь собственные более эффективные контроллеры. Режимы обмена по каналу DMA бывают одиночными и множественными. При *одиночном режиме* (*Single Word DMA*) устройство для переда-

чи каждого слова вырабатывает сигнал запроса DMARQ и сбрасывает его по сигналу DMACK#, подтверждающему цикл обмена. При *множественном режиме* (*Multiword DMA*) на сигнал DMARQ хост отвечает потоком циклов, сопровождаемых сигналами DMACK#. Если устройство не справляется с потоком, оно может приостановить его снятием сигнала DMARQ, а по готовности установить его снова. Множественный режим позволяет развить более высокую скорость передачи.

Новейшее достижение — режим *Ultra DMA*, позволяющий достигнуть скорости передачи 33 Мбайт/с и обеспечить достоверность передачи, чего не делалось ни в PIO, ни в стандартных режимах DMA (а зря!). Стандартом ATA-4 определено 3 режима Ultra DMA (0, 1 и 2), выбор режима осуществляется командой *Set Features*. В режимах Ultra DMA сигналы DMARQ и DACK# сохраняют свое назначение, а вот смысл сигналов DIOR#, DIOW# и IORDY на время передачи пакета (Ultra DMA Burst) существенно меняется (см. 6.1). В пакете данные на шине сопровождаются стробом, генерируемым источником данных, причем для синхронизации используются оба перепада сигналов. Это позволяет повысить пропускную способность шины, не увеличивая частоту переключений сигналов сверх 8,33 c^{-1} (этот предел для обычного кабеля достигается в режиме PIO Mode 4 и Multiword DMA Mode 2). Каждое переданное слово участвует в подсчете CRC-кода, который передается в конце пакета. Подсчет ведется и источником данных, и приемником. При несовпадении принятого и ожидаемого кода фиксируется ошибка передачи. Передача в пакете может приостанавливаться, если приемник снимет сигнал готовности (DDMARDY# или HDMARDY#). Передача пакета может прекращаться по инициативе устройства (снятием сигнала) или хоста (сигналом STOP). Противоположная сторона должна подтвердить окончание цикла сигналом STOP или DMARQ соответственно.

Правильный *выбор режима обмена* обеспечивает надежность и производительность. Все устройства поддерживают режим PIO Mode 0, в котором считывается блок параметров идентификации. В блоке имеются поля, описывающие режим обмена по умолчанию и более эффективные режимы обмена, поддерживаемые устройством. Командой *Set Features*

можно изменить параметры режима. Иногда накопитель не обеспечивает надежной передачи данных в заявленном высокоскоростном режиме. Если данные начинают пропадать, первым делом следует понизить режим обмена.

Параметры стандартных режимов обмена по DMA приведены в табл. 6.11.

Таблица 6.11. Параметры циклов DMA для интерфейса ATA

Режим	Минимальное время цикла, нс	Скорость передачи, Мбайт/с
Single word DMA Mode 0	960	2,08
Single word DMA Mode 1	480	4,16
Single word DMA Mode 2	240	8,33
Multiword DMA Mode 0	480	4,12
Multiword DMA Mode 1	150	13,3
Multiword DMA Mode 2	120	16,6
Ultra DMA Mode 0	240*	16,6
Ultra DMA Mode 1	160*	25
Ultra DMA Mode 2	120*	33

* В пакете данных режима Ultra DMA за каждый такт передаются два слова данных, один по фронту синхронизирующего сигнала, другой — по спаду.

BIOS определяет режим обмена с каждым устройством с учетом ограничений, заданных в Setup. Старые диски, не сообщающие своих параметров, могут не работать со старшими режимами PIO. На одном шлейфе (канале ATA) могут присутствовать устройства с разным режимом обмена — спецификация это допускает. Однако реально могут возникать аппаратные или программные ограничения. Некоторые чипсеты не позволяют независимо программировать режим обмена для устройств канала. В таком случае при подключении двух разных устройств (например, PIO Mode 1 и 3) обмен с обоими устройствами будет происходить со скоростью меньшего (PIO Mode 1). Поэтому не рекомендуется к одному каналу ATA (порту IDE, что то же самое) подключать быстрый винчестер и медленный CD-ROM. Иногда завязка режимов обмена двух устройств обусловлена ограни-

ченным набором параметров конфигурации в BIOS. Быстрые режимы множественного обмена по DMA реализуются только драйверами ОС. «Глупый» драйвер может попытаться навязать медленный режим обоим устройствам канала, так что смешивать разные устройства не стоит и по этой причине.

Протокол обмена *PIO* хорош только для *однозадачных ОС*. Для *многозадачных ОС* больший интерес представляет обмен по DMA, если, конечно, поддерживаемый режим обеспечивает приемлемую скорость обмена (сравните табл. 6.10 и 6.11).

6.6. Средства многозадачности (ATA-4)

Главным недостатком интерфейса ATA в многозадачных системах является то, что когда одно устройство на шине исполняет команду, другое использоваться не может. В этом ATA существенно уступает SCSI, где устройства на время длительной внутренней операции могут освобождать шину и имеется эффективный механизм организации очередей процессов ввода/вывода. Спецификация ATA-4 определяет возможности параллельного выполнения команд обоими устройствами и создания очередей.

Перекрытие команд — Overlapped Feature — позволяет устройству, занятому длительной внутренней операцией, освободить шину. Для этого устройство должно сбросить биты *DRQ* и *BSY* регистра состояния. По окончании выполнения операции устройство устанавливает бит *SERV* в своем регистре состояния. Если хост намерен использовать шину для обращения к другому устройству, он должен запретить прерывания от устройства, установив бит *nIEN* в регистре управления. После того как хост вернется к обслуживанию устройства, он должен послать ему команду *NOP* с подкомандой 01 для получения информации о состоянии. При этом он может разрешить прерывание, которое произойдет по готовности устройства к продолжению. Обнаружив установленный бит *SERV*, хост посылает команду *Service*, которая вызовет продолжение исполнения команды, во время которой шина освобождалась. Принятый механизм продолжения менее эффективен, чем в SCSI, — он требует привлечения хоста для обнаружения готовности устройства. Пере-

крытие допускается только для команд *NOP* (с подкомандой 01), *Packet*, *Read DMA Queued*, *Service* и *Write DMA Queued*.

Устройства могут поддерживать *очереди команд*, но только для команд, допускающих перекрытие (эти свойства тесно связаны). Если при наличии команд в очереди устройство получает команду, не входящую в этот список, команда отвергается (с соответствующим битом в регистре ошибок) и очередь сбрасывается. Глубина очереди, поддерживаемой устройством, сообщается в блоке параметров идентификации. Команды ставятся в очередь с уникальным идентификатором-тегом, который передается через регистр счетчика секторов. После исполнения команды *Service* значение тега для обслуживаемой команды считывается из того же регистра, что позволяет ее идентифицировать. Если устройство получает команду со значением тега, которое уже присутствует в очереди, и новая, и старая команды отвергаются (состояние не определено). При возникновении любой ошибки вся очередь сбрасывается. Следующая команда в очередь посылается лишь при освобожденной шине. Перед посылкой хост должен запретить прерывания, а разрешить их может только после посылки новой команды. Если для выполнения команды устройству не нужно освобождать шину, команда будет выполнена немедленно, а находящиеся в очереди команды, освободившие шину, будут выполнены позже. Очереди SCSI более эффективны как по набору команд (в очередь ставятся процессы, которые могут представлять собой цепочки команд), так и по гибкости управления. SCSI также обеспечивает независимость исполнения одних команд очереди от результатов выполнения других.

6.7. Пакетный интерфейс ATAPI

Для подключения к интерфейсу ATA накопителей CD-ROM и стриммеров (а также других устройств) набора регистров и системы команд ATA недостаточно. Для них существует аппаратно-программный интерфейс *ATAPI* (ATA Package Interface — пакетный интерфейс ATA). *Устройство ATAPI* поддерживает минимальный набор команд ATA, который

неограниченно расширяется 16-байтным командным *пакетом*, посылаемым хост-контроллером в регистр данных устройства по команде *Packet*. Структура командного пакета пришла от SCSI, что обеспечивает схожесть драйверов для устройств со SCSI и ATAPI. Классификация устройств совпадает с принятой в SCSI (см. табл. 5.12), класс устройства сообщается им в начале блока параметров идентификации.

Интерфейс ATAPI может использоваться с неинтеллектуальными адаптерами ATA, поскольку для хост-адаптера поддержка ATAPI может выполняться чисто программно. Сложные контроллеры ATA, имеющие кэш-память и собственный процессор, неориентированные на интерфейс ATAPI, могут не догадаться, что в регистр данных устройства кроме 512-байтных блоков данных можно записывать 16-байтный с командным пакетом. Устройства ATAPI имеют следующие особенности:

- Команду *Identify Device* они должны отвергать, чтобы хост не пытался к ним обратиться как к ATA-устройствам. Для идентификации устройств ATAPI предназначена специальная команда *Identify Packet Device*, а блок параметров, сообщаемых устройством, трактуется иначе (табл. 6.12).
- Для программного сброса устройства ATAPI предназначена команда *Device Reset*, которую устройства ATA отвергают. Программный сброс через регистр управления не прекращает выполнения команды *Packet*.
- Специфические команды вместе с необходимыми параметрами передаются по команде *Packet*, код которой является недействительным для устройств ATA.

Таблица 6.12. Идентификация устройств ATAPI

Слово	F/V	Назначение
0	F	*Общая конфигурационная информация* Биты [15:14] = 10 для всех устройств ATAPI Бит 13 – зарезервирован Биты 12–8 – набор поддерживаемых команд (класс устройства) Бит 7: 1 – устройство со сменным носителем Бит 6–5: 00 – устройство установит *DRQ* в течение 3 мс после получения команды *Packet*,

Слово	F/V	Назначение
		01 — устройство введет INTRQ по установке *DRQ* после получения команды *Packet*, 10 — устройство установит *DRQ* в течение 50 мкс после получения команды *Packet*, 11 — зарезервировано Биты 4–2 — зарезервированы Биты 1–0 — длина командного пакета: 00 — 12 байт, 01 — 16 байт, 1x — зарезервировано
1–9	R	Зарезервировано
10–19	F	*Серийный номер* (20 символов ASCII), если слово 10 не нулевое; иначе — специфичная информация
20–22	R	Зарезервировано
23–26	F	*Версия встроенного ПО* (8 символов ASCII), если слово 23 не нулевое; иначе — специфичная информация
27–46	F	*Номер модели* (40 символов ASCII), если слово 27 не нулевое; иначе — специфичная информация
47–48	R	Зарезервировано
49		*Возможности:*
	F	Бит 15: 1 — поддержка чередующегося (interleaved) DMA
	F	Бит 14: 1 — поддержка очередей команд
	F	Бит 13: 1 — поддержка перекрывающихся команд
	F	Бит 12: 1 — требуется программный сброс (устарело)
	F	Бит 11: 1 — IORDY поддерживается (для PIO Mode 3 и выше — обязательно), 0 — IORDY не обязано поддерживаться
	F	Бит 10: 1 — сигнал IORDY может быть запрещен командой *Set Features*
	F	Бит 9: 1 — поддержка LBA
	F	Бит 8: 1 — поддержка DMA
	X	Биты 7–0 специфичны

Слово	F/V	Назначение
50	R	Зарезервировано
51	F X	Биты 15–8: номер PIO Mode Биты 7–0 специфичны
52	R	Зарезервировано
53	R F F V	Биты 15–3 зарезервированы Бит 2: 1 — поля в слове 88 действительны, 0 — нет Бит 1: 1 — поля в словах 64–70 действительны, 0 — нет Устройство, поддерживающее PIO Mode 3 и старше или Multiword DMA Mode 1 и выше, должно использовать эти поля Бит 0: 1 — поля в словах 54–58 действительны, 0 — поля могут быть действительны
54–62	R	Зарезервировано
63	R R V	*Режим множественного обмена Multiword DMA*: Биты 15–11 зарезервированы Биты 10–8: активный режим: бит 8=1 — Mode 0, бит 9=1 — Mode 1 и т. д. Единичное значение может иметь только один бит Биты 7–3 зарезервированы Биты 2–0: поддерживаемые режимы: бит 0=1 — Mode 0, бит 1=1 — Mode 1 и т. д.
64	R F	Биты 15–8 зарезервированы Биты 7–0 — *поддерживаемые прогрессивные режимы программного обмена (Advanced PIO)*: бит 0=1 — PIO Mode 3, бит 1=1 — PIO Mode 4, биты 2–7 — зарезервированы
65	F	*Минимальная длительность цикла передачи множественного DMA* (в наносекундах)
66	F	*Рекомендованная длительность цикла передачи множественного DMA* (в наносекундах). При многосекторной передаче обеспечивает оптимальную скорость, при которой устройство не тормозит поток снятием запроса DMARQ

Слово	F/V	Назначение
67	F	*Минимальная длительность цикла передачи PIO без использования сигнала готовности* (в наносекундах)
68	F	*Минимальная длительность цикла передачи PIO с использованием сигнала готовности* (в наносекундах)
69–70	R	Зарезервированы для поддержки перекрытия команд и очередей
71	F	Типовое время от получения команды *Packet* до освобождения шины (в наносекундах)
72	F	Типовое время от получения команды *Service* до сброса *BSY* (в наносекундах)
73–74	R	Зарезервировано
75	F	*Глубина очереди команд*: биты 15–5 — зарезервированы, биты 4–0 -- максимальная глубина очереди
76–79	R	Зарезервировано
80	F	*Номер основной версии* интерфейса (если не 0000h или FFFFh): 1 – ATA-1, 2 – ATA-2 и т. д.
81	F	*Младший номер версии* интерфейса (если не 0000h или FFFFh)
82	F	Поддержка команд и свойств (если слова 82 и 83 не равны 0000h или FFFFh): Бит 15 – не используется Бит 14 – команды *Nop* Бит 13 – команды *Read Buffer* Бит 12 – команды *Write Buffer* Бит 11 – не используется Бит 10 – ограничение доступного пространства Бит 9 – команда *Device Reset* Бит 8 – вырабатывание прерывания во время команды *Service* Бит 7 – прерывание по освобождению шины Бит 6 – кэширование с упреждающим чтением Бит 5 – кэширование записи Бит 4 – команда *Packet* Бит 3 – управление энергопотреблением Бит 2 – смена носителя Бит 1 – команды *Security* Бит 0 – *SMART*

Таблица 6.12 (продолжение)

Слово	F/V	Назначение
83	F	*Поддержка команд и свойств* (как и слово 82): Бит 15 = 0 Бит 14 = 1 Биты 13–5 — зарезервированы Бит 4 — уведомление о смене носителя Биты 3–1 — зарезервированы Бит 0 — команда *Download Microcode*
84	F	*Поддержка команд и свойств* (если слова 82, 83 и 84 не равны 0000h или FFFFh): Бит 15 = 0 Бит 14 = 1 Биты 13–0 — зарезервированы
85–87	F	*Разрешенные команды и свойства* — поля и правила аналогичны словам 82–84
88	R R V	*Режим Ultra DMA*: Биты 15–11 — зарезервированы Биты 10–8 — активный режим Ultra DMA: бит 8=1 — Mode 0, бит 9=1 — Mode 1 и т. д. Единичное значение может иметь только один бит Биты 7–3 — зарезервированы Биты 2–0 — поддерживаемые режимы одиночного обмена DMA: бит 0=1 — Mode 0, бит 1=1 — Mode 1 и т. д.
89–126	R	Зарезервировано
127	R F	Биты 15–2 — зарезервированы Биты 1–0 — уведомление о смене носителя: 00 — не поддерживается, 01 — поддерживается, 1x — зарезервированы
128	V	*Состояние защиты:* Биты 15–9 — зарезервированы Бит 8 — уровень защиты: 0 — высокий, 1 — максимальный Биты 7–6 — зарезервированы Бит 5: 1 — поддержка расширенного защитного стирания Бит 4: 1 — счетчик попыток иссяк Бит 3: 1 — команды защиты блокированы

Слово	F/V	Назначение
		Бит 2: 1 — устройство заблокировано Бит 1: 1 — защита разрешена Бит 0: 1 — защита поддерживается
129–159	X	Специфично
160–255	R	Зарезервировано

При подаче команды ***Packet*** регистр свойств ***FR*** содержит признаки команды:

- Бит 0 — ***DMA*** — является указанием на использование DMA или Ultra DMA для обмена данными.
- Бит 1 — ***OVL*** — является признаком возможности перекрывающегося выполнения.

Регистры ***CH*** и ***CL*** содержат лимит счетчика байт данных, передаваемых по каждому введению ***DRQ*** в режиме PIO. Если команда не предусматривает обмена данными, поле игнорируется. Если общее число требуемых байт данных превышает лимит, значение лимита задается четным. Если оно равно или меньше лимита, лимит нечетен. Значение лимита FFFFh воспринимается устройством как FFFEh. Лимит счетчика, как и бит DMA, к передаче пакета отношения не имеет.

Регистр ***DH*** используется только для выбора устройства. Для устройств, поддерживающих очереди команд, биты [7:3] регистра ***SC*** содержат тег (***Tag***).

После подачи команды регистр ***SC*** кроме тега будет содержать следующие биты:

- Бит 2 — ***REL*** (Release) — признак освобождения шины (для перекрывающихся команд).
- Бит 1 — ***I/O*** (Input/Output) — указатель направления передачи данных (0 — передача к устройству, 1 — к хосту).
- Бит 0 — ***C/D*** (Command/Data) — признак передачи команды (1) или данных (0).

В *регистре состояния* некоторые биты получают новое назначение:

- Бит 5 — ***DMRDY*** (DMA Ready) — при ***DRQ***=1 используется как признак готовности к обмену в режиме (Ultra)

DMA. При *DRQ*=0 является признаком отказа устройства *DF* (Device Fault).

- Бит 4 — *SERV* (Service) — признак готовности к обслуживанию команды, освободившей шину.
- Биты 1, 2 не используются.
- Бит 0 — *CHK* (Check) — признак ошибки.

Остальные биты сохранили обычное назначение.

После подачи команды *Packet* хост определяет состояние устройства, прочитав его регистры. Возможны следующие варианты:

- *Ожидание команды Packet*: в регистре *SC* биты *C/D*=1, *I/O*=0, *REL*=0, поле *Tag* содержит ранее установленное значение. В регистре состояния *BSY*=0, *DMRDY*=0, *CHK*=0, *DRQ*=1, бит *SERV* может указывать на наличие команды, ожидающей обслуживания.
- *Передача данных*: в регистре *SC* биты *C/D*=0, *REL*=0, *I/O* указывает на направление передачи. Регистры *CH*, *CL* при использовании PIO содержат счетчик байт данных. В регистре состояния *BSY*=0, *DRQ*=1, *CHK*=0, *DMRDY* указывает на использование DMA, бит *SERV* может указывать на наличие команды, ожидающей обслуживания.
- *Освобождение шины*: в регистре *SC* биты *C/D*=0, *I/O*=0, *REL*=1, поле *Tag* содержит тег. В регистре состояния *BSY*=0, *DMRDY*=0, *CHK*=0, *DRQ*=0, бит *SERV* может указывать на наличие команды, ожидающей обслуживания.
- *Запрос обслуживания*: в регистре *SC* биты *C/D*=0, *I/O*=0, *REL*=1, поле *Tag* содержит тег. В регистре состояния *BSY*=0, *DMRDY*=0, *CHK*=0, *DRQ*=0, бит *SERV*=1.
- *Успешное завершение*: в регистре *SC* биты *C/D*=1, *I/O*=1, *REL*=0, поле *Tag* содержит тег. В регистре состояния *BSY*=0, *DRDY*=1, *CHK*=0, *DRQ*=0, бит *SERV* может указывать на наличие команды, ожидающей обслуживания.
- *Завершение с ошибкой* (только после передачи последнего байта пакета): в регистре *SC* биты *C/D*=1, *I/O*=1, *REL*=0, поле *Tag* содержит тег. В регистре состояния *BSY*=0, *DRDY*=1, *DRQ*=0, *DF*=1 при отказе устройства, *CHK*=1, если

регистр ошибок содержит код ошибки, бит *SERV* может указывать на наличие команды, ожидающей обслуживания. В регистре ошибок биты 0 и 1 трактуются в зависимости от команды, бит 2 — признак отвергнутой команды (*ABRT*), биты [7:4] могут содержать уточненное состояние.

Структуру командного пакета см. в 5.5. При любой длине блока дескрипторов, определяемой кодом команды (нулевой байт пакета), передаваемый пакет имеет длину 16 байт, но используется только указанное количество байт. Систему команд и структуру пакетов стандарт ATA/ATAPI-4 не описывает, но для каждого класса устройств существует стандартизованный набор команд с определенной структурой пакетов.

6.8. Адаптеры шины АТА и категории устройств IDE

Простейший *адаптер АТА* содержит только буферы сигналов шины и дешифратор зоны адресов. Все регистры контроллера и схемы кодирования размещены в самом устройстве IDE. Шина АТА требует выделения системных ресурсов — двух областей портов ввода/вывода и линии прерывания; дополнительно может использоваться канал DMA. Интерфейсу АТА первого канала выделили ресурсы, ранее использовавшиеся контроллером жестких дисков. Второму каналу назначили ресурсы альтернативного контроллера жестких дисков. Позже определили ресурсы еще для двух каналов (табл. 6.13). Традиционному контроллеру жестких дисков выделялся канал DMA3, но он является 8-битным, в то время как шина АТА требует 16-битного канала DMA. Производительности стандартных каналов DMA для шины АТА явно недостаточно. На системных платах с шиной PCI стандартом является установка *двухканального адаптера*, занимающего ресурсы каналов 1 и 2. Этот адаптер обеспечивает шинам АТА 16-битные каналы DMA, поддерживающие высокоскоростные режимы обмена (см. табл. 6.11). В идеальном варианте двухканальные контроллеры имеют шины, полностью изолированные друг от друга буферными и логическими схемами. В самом дешевом варианте они ис-

пользуют общие буферы для линий данных и управляющих сигналов и отдельные только для некоторых сугубо индивидуальных сигналов. С точки зрения логики, здесь все в порядке, но следует учитывать нагрузочную способность (влияние паразитных параметров): *суммарная длина* обоих шлейфов не должна превышать 46 см, а *суммарная емкость* каждой линии со всеми устройствами не должна превышать 35 пФ. Иначе на высокоскоростных режимах обмена возможны *неконтролируемые искажения* передаваемых данных.

Таблица 6.13. Системные ресурсы каналов ATA

Канал	CS0	CS1	IRQ
1	1F0h–1F7h	3F6h–3F7h	14
2	170h–177h	376h–377h	15 или 10
3	1E8h–1Efh	3EEh–3Efh	12 или 11
4	168h–16Fh	36Eh–36Fh	10 или 9

Поскольку скорость программного обмена задается хост-адаптером, интересно индивидуальное программирование PIO Mode для каждого канала/устройства. Ряд чипсетов этого не допускает и при инициализации назначает общий минимальный режим. В результате подключение «тихоходного» устройства замедляет обмен быстрого соседнего устройства.

Адаптеры ATA часто размещают на звуковых картах для подключения накопителей CD-ROM. По умолчанию им назначают ресурсы каналов 3 или 4. К этим каналам можно подключать винчестеры, но будет ли их там искать BIOS во время POST — вопрос. Современные версии BIOS позволяют хранить конфигурационные параметры четырех жестких дисков, более старые версии — двух. Четыре канала ATA физически позволяют подключить до восьми накопителей, но работа с ними лимитирована программными ограничениями.

Расширенные адаптеры могут иметь аппаратные средства поддержки высокоэффективных режимов передачи (PIO, DMA, Bus Master), буферные регистры записи и собственную кэш-память. Сложные адаптеры аппаратно поддерживают «зеркальные» диски. Некоторые адаптеры позволяют соединять несколько физических дисков в один логический

на уровне вызовов BIOS (в настоящее время актуальнее обратная задача).

Существуют гибридные адаптеры для подключения ATA HDD к шинам XT и MCA или, например, к LPT-порту.

Интерфейс ATA позволяет подключать *устройства различных категорий*, отличающихся «уровнем интеллекта» встроенного контроллера. Первые дисковые накопители IDE относятся к категории неинтеллектуальных устройств — *Non-Intelligent IDE*. Они не выполняли трансляцию нумерации секторов — их логические параметры совпадали с физическими. Команды идентификации устройства и установки параметров не выполнялись. Дефектные блоки, отмеченные в заводском списке, были видны пользователю. Низкоуровневое форматирование выполнялось непосредственно по команде, так что неудачное форматирование могло понизить производительность из-за нарушения оптимальных установок чередования и смещения.

Более интеллектуальные устройства — *Intelligent ATA IDE*. Они способны выполнять расширенные ATA-команды — идентификацию устройства и установку параметров. Поддерживается трансляция физических параметров в логические. Дефектные сектора скрыты от пользователя (до исчерпания резерва). Низкоуровневое форматирование возможно только при установке логической геометрии, совпадающей с физической. Однако форматирование опять-таки «сносит» заводскую оптимизацию. Для ускорения обмена эти устройства поддерживают блочные режимы передачи *Read Multiple* и *Write Multiple*, а также высокоскоростные режимы обмена PIO и DMA.

К следующей категории относятся устройства с зонным форматом записи — *Intelligent Zoned Recording IDE*. Поскольку они имеют различное количество секторов на разных треках (для повышения плотности хранения), трансляция геометрии является для них обязательной (спецификация ATA не предусматривает сообщения устройством способа разбиения на зоны и формата каждой зоны, так что обращаться к ним можно только по логическому трехмерному (CHS) или линейному (LBA) адресу).

Устройства IDE отличаются также по интеллектуальности контроллера: автоматический мониторинг внутренних параметров (SMART), температурная коррекция системы позиционирования, поддержка управления энергопотреблением и различные усовершенствования, направленные на повышение производительности.

6.9. Конфигурирование устройств

Устройства ATA перед подключением к шине должны быть корректно сконфигурированы. Конфигурирование подразумевает выбор типа интерфейса и определение адреса устройства. Тип интерфейса — XT или AT — определяется моделью накопителя. В изделиях фирмы Seagate тип обозначается последней буквой в шифре модели: A — ATA (16 бит), X — для XT (8 бит), а сочетание AX означает возможность выбора AT/XT с помощью джампера.

Существует два способа задания адреса устройства — с помощью кабельной выборки или явным заданием адреса на каждом из устройств. Режим *кабельной выборки* включается перемычкой *CS* (Cable Select — кабельная выборка). В этом случае оба устройства на шине конфигурируются одинаково — в режим *CS*, а адрес устройства определяется его положением на специальном кабеле-шлейфе (рис. 6.2). Кабельная выборка будет работать, если она поддерживается и задана на всех устройствах канала, включая хост-адаптер, который обеспечивает заземление контакта 28. При этом способе задания адресов исключается синхронизация шпинделей накопителей (актуально в RAID-массивах) через тот же провод контакта 28. Кабельная выборка применяется редко. Ее условное преимущество — унификация конфигурирования устройств, а недостаток — привязка физического положения устройств к кабелю — ведущее устройство должно быть ближе к адаптеру, чем ведомое. Возможно подключение адаптера к среднему разъему, а устройств — к крайним, но это не всегда удобно.

Более распространен режим явной адресации, при котором используется обычный «прямой» кабель (см. рис. 6.1). В этом случае перемычка *CS* не устанавливается, а адрес устрой-

ства задается перемычками, состав которых варьируется. В принципе, достаточно лишь указать устройству его номер (0/1), но в устройствах, разработанных до стандарта ATA, ведущему устройству «подсказывали» о наличии ведомого (по интерфейсу ATA оно могло бы это определить само по сигналу *DASP#*). Итак, на устройствах IDE можно увидеть следующие джамперы:

- *M/S* (Master/Slave — ведущее/ведомое) — переключатель адреса. Если на шине присутствует одно устройство, оно должно быть сконфигурировано как ведущее. Если на шине два устройства — одно должно быть ведущим, другое — ведомым. Иногда джампер обозначается как «*C/D*» (диск C:/диск D:), но для второго канала IDE такое название некорректно. Когда появились первые IDE-диски емкостью 1 Гбайт, для преодоления барьера в 504 Мбайт некоторые модели допускали конфигурирование в виде двух устройств (0 и 1) половинной емкости. В таком режиме на их IDE-шлейф второе физическое устройство подключать нельзя.
- *SP* (Slave Present), *DSP* (Drive Slave Present), «Master but Slave is not ATA-compatible», «Master but Slave uses only PDIAG-signal» — устанавливается на ведущем устройстве для указания на присутствие ведомого. Если переключатель установлен, а ведомое устройство не подключено, POST даст сообщение об ошибке. Джампер применяется для дисков, не использующих сигнал *DASP#*.
- *ACT* (Drive Active) — устанавливается на Master (встречается редко).

Для полностью *ATA-совместимых* дисков перемычка ставится только на ведомое устройство, а его присутствие ведущее определит автоматически.

Разобраться с джамперами старых устройств трудно, если нет документации. У современных устройств лишние джамперы упразднили, а существующие комментируются на наклейке-шильдике. Если джамперы устанавливаются рядом с интерфейсным разъемом, вероятно, они расположены в соответствии со стандартом ATA (рис. 6.4). Здесь буквой k обозначены позиции ключевых (пропущенных) выводов, контакты 1–40 ис-

пользуются для интерфейса, а контакты A–H – для установки джамперов (табл. 6.14).

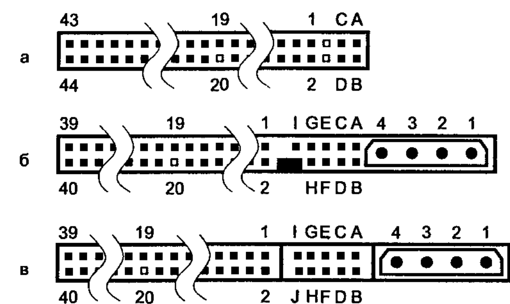


Рис. 6.4. Джамперы на разъемах интерфейса ATA:
а — SFF8212, *б* — SFF8057, *в* — SFF8058

Таблица 6.14. Установка джамперов

Выборка	SFF8212	SFF8057	SFF8058
Используемые контакты	A...D	E...H	A...F
Кабельная выборка	B–D	E–F	A–B
Master	–	G–H	E–F
Master при наличии Slave	–	G–H, E–F	E–F
Slave	A-B	–	C–D



Следует учитывать, что *перестановка джамперов* часто воспринимается устройством только *по включении питания*. Кроме того, установка на один шлейф двух разнотипных не-ATA-устройств часто невозможна.

7. Последовательные шины

Последовательные шины позволяют объединять множество устройств, используя всего 1–2 пары проводов. При этом достигается пропускная способность от 100 кбит/с для шины ACCESS.Bus до 400 Мбит/с у FireWire. Функциональные возможности этих шин гораздо шире, чем у традиционных интерфейсов локальных сетей — USB и FireWire способны передавать изохронный трафик аудио- и видеоданных.

7.1. Шина USB

USB (Universal Serial Bus — универсальная последовательная шина) является промышленным стандартом расширения архитектуры PC, ориентированным на интеграцию с телефонией и устройствами бытовой электроники. Версия 1.0 была опубликована в январе 1996 года. Архитектура USB определяется следующими критериями:

- Легко реализуемое расширение периферии PC.
- Дешевое решение, поддерживающее скорость передачи до 12 Мбит/с.
- Полная поддержка в реальном времени передачи аудио- и (сжатых) видеоданных.
- Гибкость протокола смешанной передачи изохронных данных и асинхронных сообщений.
- Интеграция с выпускаемыми устройствами.
- Доступность в PC всех конфигураций и размеров.
- Обеспечение стандартного интерфейса, способного быстро завоевать рынок.
- Создание новых классов устройств, расширяющих PC.

С точки зрения конечного пользователя, привлекательны следующие черты USB:

- Простота кабельной системы и подключений.

- Скрытие подробностей электрического подключения от конечного пользователя.
- Самоидентифицирующиеся ПУ, автоматическая связь устройств с драйверами и конфигурирование.
- Возможность динамического подключения и конфигурирования ПУ.

С середины 1996 года выпускаются PC со встроенным контроллером USB, реализуемым чипсетом. Ожидается появление модемов, клавиатур, сканеров, динамиков и других устройств ввода/вывода с поддержкой USB, а также мониторов с USB-адаптерами — они будут играть роль хабов для подключения других устройств.

7.1.1. Структура USB

USB обеспечивает одновременный обмен данными между *хост-компьютером* и множеством *периферийных устройств (ПУ)*. Распределение пропускной способности шины между ПУ планируется хостом и реализуется им с помощью посылки маркеров. Шина позволяет подключать, конфигурировать, использовать и отключать устройства во время работы хоста и самих устройств.

Ниже приводится авторский вариант перевода терминов из спецификации «Universal Serial Bus Specification. Revision 1.0, January 15, 1996», опубликованной Compaq, DEC, IBM, Intel, Microsoft, NEC и Northern Telecom. Более подробную и оперативную информацию можно найти по адресу: http://www.usb.org.

Устройства (*Device*) USB могут являться хабами, функциями или их комбинацией. *Хаб* (Hub) обеспечивает дополнительные точки подключения устройств к шине. ***Функции*** (***Function***) USB предоставляют системе дополнительные возможности, например подключение к ISDN, цифровой джойстик, акустические колонки с цифровым интерфейсом и т. п. Устройство USB должно иметь интерфейс USB, обеспечивающий полную поддержку протокола USB, выполнение стандартных операций (конфигурирование и сброс) и предоставление информации, описывающей устройство. Мно-

гие устройства, подключаемые к USB, имеют в своем составе и хаб, и функции. Работой всей системы USB управляет *хост-контроллер* (*Host Controller*), являющийся программно-аппаратной подсистемой хост-компьютера.

Физическое соединение устройств осуществляется по топологии многоярусной *звезды.* Центром каждой звезды является *хаб*, каждый кабельный сегмент соединяет две точки — хаб с другим хабом или с функцией. В системе имеется один (и только один) *хост-контроллер*, расположенный в вершине пирамиды устройств и хабов. Хост-контроллер интегрируется с *корневым хабом* (*Root Hub*), обеспечивающим одну или несколько точек подключения — *портов.* Контроллер USB, входящий в состав чипсетов, обычно имеет встроенный двухпортовый хаб. Логически устройство, подключенное к любому хабу USB и сконфигурированное (см. ниже), может рассматриваться как непосредственно подключенное к хост-контроллеру.

Функции представляют собой устройства, способные передавать или принимать данные или управляющую информацию по шине. Типично функции представляют собой отдельные ПУ с кабелем, подключаемым к порту хаба. Физически в одном корпусе может быть несколько функций со встроенным хабом, обеспечивающим их подключение к одному порту. Эти комбинированные устройства для хоста являются хабами с постоянно подключенными устройствами-функциями.

Каждая функция предоставляет конфигурационную информацию, описывающую возможности ПУ и требования к ресурсам. Перед использованием функция должна быть сконфигурирована хостом — ей должна быть выделена полоса в канале и выбраны опции конфигурации.

Примерами функций являются:

- Указатели — мышь, планшет, световое перо.
- Устройства ввода — клавиатура или сканер.
- Устройство вывода — принтер, звуковые колонки (цифровые).
- Телефонный адаптер ISDN.

Хаб — ключевой элемент системы PnP в архитектуре USB. Хаб является кабельным концентратором. Точки подключения называются *портами* хаба. Каждый хаб преобразует одну точку подключения в их множество. Архитектура допускает соединение нескольких хабов.

У каждого хаба имеется один *восходящий порт* (*Upstream Port*), предназначенный для подключения к хосту или хабу верхнего уровня. Остальные порты являются *нисходящими* (*Downstream Ports*), предназначенными для подключения функций или хабов нижнего уровня. Хаб может распознать подключение устройств к портам или отключение от них и управлять подачей питания на их сегменты. Каждый из портов может быть разрешен или запрещен и сконфигурирован на полную или ограниченную скорость обмена. Хаб обеспечивает изоляцию сегментов с низкой скоростью от высокоскоростных.

Хабы могут управлять подачей питания на нисходящие порты; предусматривается установка ограничения на ток, потребляемый каждым портом.

Рис. 7.1. Взаимодействие компонентов USB

Система USB разделяется на три уровня с определенными правилами взаимодействия. Устройство USB содержит интерфейсную часть, часть устройства и функциональную часть. Хост тоже делится на три части — интерфейсную, системную и ПО устройства. Каждая часть отвечает только за определенный круг задач, логическое и реальное взаимодействие между ними иллюстрирует рис. 7.1.

В рассматриваемую структуру входят следующие элементы:

- *Физическое устройство USB* — устройство на шине, выполняющее функции, интересующие конечного пользователя.
- *Client SW* — ПО, соответствующее конкретному устройству, исполняемое на хост-компьютере. Может являться составной частью ОС или специальным продуктом.
- *USB System SW* — системная поддержка USB, независимая от конкретных устройств и клиентского ПО.
- *USB Host Controller* — аппаратные и программные средства для подключения устройств USB к хост-компьютеру.

Физический интерфейс

Стандарт USB определяет электрические и механические спецификации шины.

Информационные сигналы и питающее напряжение 5 В передаются по четырехпроводному кабелю. Используется дифференциальный способ передачи сигналов D+ и D– по двум проводам. Уровни сигналов передатчиков в статическом режиме должны быть ниже 0,3 В (низкий уровень) или выше 2,8 В (высокий уровень). Приемники выдерживают входное напряжение в пределах – 0,5...+3,8 В. Передатчики должны уметь переходить в высокоимпедансное состояние для двунаправленной полудуплексной передачи по одной паре проводов.

Передача по двум проводам в USB не ограничивается дифференциальными сигналами. Кроме дифференциального приемника каждое устройство имеет линейные приемники сигналов D+ и D–, а передатчики этих линий управляются индивидуально. Это позволяет различать более двух состоя-

ний линии, используемых для организации аппаратного интерфейса. Состояния *Diff0* и *Diff1* определяются по разности потенциалов на линиях D+ и D– более 200 мВ при условии, что на одной из них потенциал выше порога срабатывания V_{SE}. Состояние, при котором на обоих входах D+ и D– присутствует низкий уровень, называется *линейным нулем* (*SE0 — Single-Ended Zero*). Интерфейс определяет следующие состояния:

- *Data J State* и *Data K State* — состояния передаваемого бита (или просто *J* и *K*), определяются через состояния *Diff0* и *Diff1*.
- *Idle State* — пауза на шине.
- *Resume State* — сигнал «пробуждения» для вывода устройства из «спящего» режима.
- *Start of Packet (SOP)* — начало пакета (переход из *Idle State* в *K*).
- *End of Packet (EOP)* — конец пакета.
- *Disconnect* — устройство отключено от порта.
- *Connect* — устройство подключено к порту.
- *Reset* — сброс устройства.

Состояния определяются сочетаниями дифференциальных и линейных сигналов; для полной и низкой скоростей состояния *Diff0* и *Diff1* имеют противоположное назначение. В декодировании состояний *Disconnect*, *Connect* и *Reset* учитывается время нахождения линий (более 2,5 мс) в определенных состояниях.

Шина имеет два режима передачи. *Полная скорость* передачи сигналов USB составляет 12 Мбит/с, *низкая* — 1,5 Мбит/с. Для полной скорости используется экранированная витая пара с импедансом 90 Ом и длиной сегмента до 5 м, для низкой — невитой неэкранированный кабель до 3 м. Низкоскоростные кабели и устройства дешевле высокоскоростных. Одна и та же система может одновременно использовать оба режима; переключение для устройств осуществляется прозрачно. Низкая скорость предназначена для работы с небольшим количеством ПУ, не требующих высокой скорости.

Скорость, используемая устройством, подключенным к конкретному порту, определяется хабом по уровням сигналов на линиях D+ и D–, смещаемых нагрузочными резисторами R2 приемопередатчиков (см. рис. 7.2 и 7.3).

Сигналы синхронизации кодируются вместе с данными по методу *NRZI* (*Non Return to Zero Invert*), его работу иллюстрирует рис. 7.4. Каждому пакету предшествует поле синхронизации *SYNC*, позволяющее приемнику настроиться на частоту передатчика.

Кабель также имеет *линии* VBus и GND для передачи питающего напряжения 5 В к устройствам. Сечение проводников выбирается в соответствии с длиной сегмента для обеспечения гарантированного уровня сигнала и питающего напряжения.

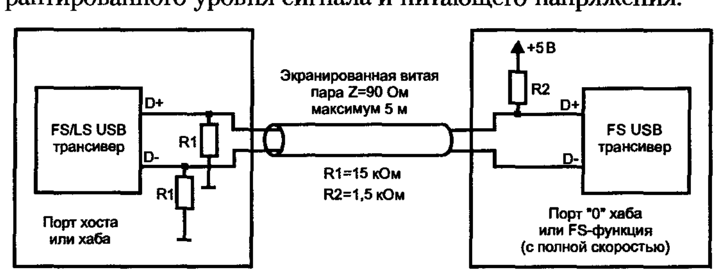


Рис. 7.2. Подключение полноскоростного устройства

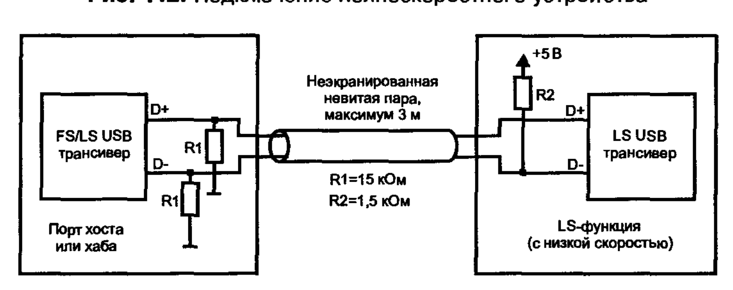


Рис. 7.3. Подключение низкоскоростного устройства

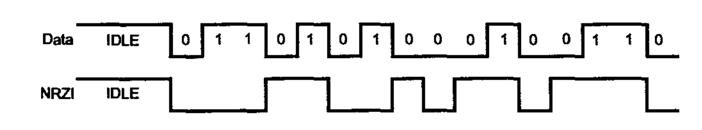


Рис. 7.4. Кодирование данных по методу NRZI

Стандарт определяет *два типа разъемов* (см. табл. 7.1 и рис. 7.5).

Таблица 7.1. Назначение выводов разъема USB

Контакт	Цепь	Контакт	Цепь
1	VBus	3	D+
2	D–	4	GND

Разъемы типа «A» применяются для подключения к хабам (*Upstream Connector*). Вилки устанавливаются на кабелях, не отсоединяемых от устройств (например, клавиатура, мышь и т. п.). Гнезда устанавливаются на нисходящих портах (*Downstream Port*) хабов.

Разъемы типа «B» (*Downstream Connector*) устанавливаются на устройствах, от которых соединительный кабель может отсоединяться (принтеры и сканеры). Ответная часть (вилка) устанавливается на соединительном кабеле, противоположный конец которого имеет вилку типа «A».

Разъемы типов «A» и «B» различаются механически (рис. 7.5), что исключает недопустимые петлевые соединения портов хабов. Четырехконтактные разъемы имеют ключи, исключающие неправильное присоединение. Конструкция разъемов обеспечивает позднее соединение и раннее отсоединение сигнальных цепей по сравнению с питающими. Для распознавания разъема USB на корпусе устройства ставится стандартное символическое обозначение.

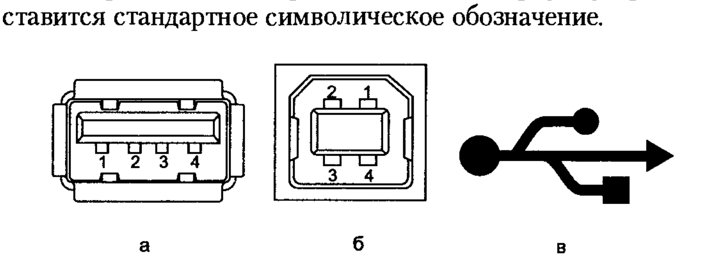


Рис. 7.5. Гнезда USB: *а* — типа «A», *б* — типа «B», *в* — символическое обозначение

Питание устройств USB возможно *от кабеля* (*Bus-Powered Devices*) или от собственного блока питания (*Self-Powered*

Devices). Хост обеспечивает питанием непосредственно подключенные к нему ПУ. Каждый хаб, в свою очередь, обеспечивает питание устройств, подключенных к его нисходящим портам. При некоторых ограничениях топологии допускается применение хабов, питающихся от шины. На рис. 7.6 приведен пример схемы соединения устройств USB. Здесь клавиатура, перо и мышь могут питаться от шины.

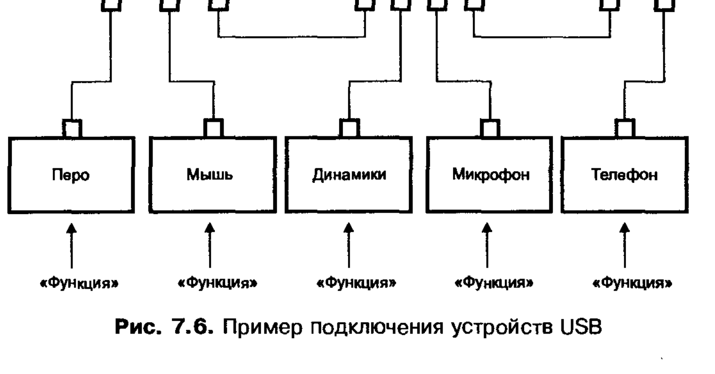


Рис. 7.6. Пример подключения устройств USB

Модель передачи данных

Каждое устройство USB представляет собой набор независимых *конечных точек* (*Endpoint*), с которыми хост-контроллер обменивается информацией. Конечные точки описываются следующими параметрами:

- требуемой частотой доступа к шине и допустимыми задержками обслуживания;
- требуемой полосой пропускания канала;
- номером точки;
- требованиями к обработке ошибок;
- максимальными размерами передаваемых и принимаемых пакетов;

- типом обмена;
- направлением обмена (для сплошного и изохронного обменов).

Каждое устройство обязательно имеет конечную точку с номером 0, используемую для инициализации, общего управления и опроса его состояния. Эта точка всегда сконфигурирована при включении питания и подключении устройства к шине. Оно поддерживает передачи типа «управление» (см. далее).

Кроме нулевой точки, устройства-функции могут иметь дополнительные точки, реализующие полезный обмен данными. Низкоскоростные устройства могут иметь до двух дополнительных точек, полноскоростные — до 16 точек ввода и 16 точек вывода (протокольное ограничение). Точки не могут быть использованы до их конфигурирования (установления согласованного с ними канала).

Каналом (*Pipe*) в USB называется модель передачи данных между хост-контроллером и конечной точкой (*Endpoint*) устройства. Имеются два типа каналов: потоки (*Stream*) и сообщения (*Message*). *Поток* доставляет данные от одного конца канала к другому, он всегда однонаправленный. Один и тот же номер конечной точки может использоваться для двух поточных каналов — ввода и вывода. Поток может реализовывать следующие типы обмена: сплошной, изохронный и прерывания. Доставка всегда идет в порядке «первым вошел — первым вышел» (FIFO); с точки зрения USB, данные потока неструктурированы. *Сообщения* имеют формат, определенный спецификацией USB. Хост посылает запрос к конечной точке, после которого передается (принимается) пакет сообщения, за которым следует пакет с информацией состояния конечной точки. Последующее сообщение нормально не может быть послано до обработки предыдущего, но при отработке ошибок возможен сброс необслуженных сообщений. Двухсторонний обмен сообщениями адресуется к одной и той же конечной точке. Для доставки сообщений используется только обмен типа «управление».

С каналами связаны характеристики, соответствующие конечной точке (полоса пропускания, тип сервиса, размер буфера

и т. п.). Каналы организуются при конфигурировании устройств USB. Для каждого включенного устройства существует канал сообщений (*Control Pipe 0*), по которому передается информация конфигурирования, управления и состояния.

Типы передачи данных

USB поддерживает как однонаправленные, так и двунаправленные режимы связи. Передача данных производится между ПО хоста и конечной точкой устройства. Устройство может иметь несколько конечных точек, связь с каждой из них (канал) устанавливается независимо.

Архитектура USB допускает четыре базовых типа передачи данных:

- *Управляющие посылки* (*Control Transfers*), используемые для конфигурирования во время подключения и в процессе работы для управления устройствами. Протокол обеспечивает гарантированную доставку данных. Длина поля данных управляющей посылки не превышает 64 байт на полной скорости и 8 байт на низкой.
- *Сплошные передачи* (*Bulk Data Transfers*) сравнительно больших пакетов без жестких требований ко времени доставки. Передачи занимают всю свободную полосу пропускания шины. Пакеты имеют поле данных размером 8, 16, 32 или 64 байт. Приоритет этих передач самый низкий, они могут приостанавливаться при большой загрузке шины. Допускаются только на полной скорости передачи.
- *Прерывания* (*Interrupt*) – короткие (до 64 байт на полной скорости, до 8 байт на низкой) передачи типа вводимых символов или координат. Прерывания имеют спонтанный характер и должны обслуживаться не медленнее, чем того требует устройство. Предел времени обслуживания устанавливается в диапазоне 1–255 мс для полной скорости и 10–255 мс – для низкой.
- *Изохронные передачи* (*Isochronous Transfers*) – непрерывные передачи в реальном времени, занимающие предварительно согласованную часть пропускной способности шины и имеющие заданную задержку доставки. В случае

обнаружения ошибки изохронные данные передаются без повтора — недействительные пакеты игнорируются. Пример — цифровая передача голоса. Пропускная способность определяется требованиями к качеству передачи, а задержка доставки может быть критичной, например, при реализации телеконференций.

Полоса пропускания шины делится между всеми установленными каналами. Выделенная полоса закрепляется за каналом, и если установление нового канала требует такой полосы, которая не вписывается в уже существующее распределение, запрос на выделение канала отвергается.

Архитектура USB предусматривает внутреннюю буферизацию всех устройств, причем чем большей полосы пропускания требует устройство, тем больше должен быть его буфер. USB должна обеспечивать обмен с такой скоростью, чтобы задержка данных в устройстве, вызванная буферизацией, не превышала нескольких миллисекунд.

Изохронные передачи классифицируются по способу синхронизации конечных точек — источников или получателей данных — с системой: различают асинхронный, синхронный и адаптивный классы устройств, каждому из которых соответствует свой тип канала USB.

Протокол

Все обмены (транзакции) по USB состоят из трех пакетов. Каждая *транзакция* планируется и начинается по инициативе контроллера, который посылает *пакет-маркер* (*Token Packet*). Он описывает тип и направление передачи, адрес устройства USB и номер конечной точки. В каждой транзакции возможен обмен только между адресуемым устройством (его конечной точкой) и хостом. Адресуемое маркером устройство распознает свой адрес и готовится к обмену. Источник данных (определенный маркером) передает пакет данных (или уведомление об отсутствии данных, предназначенных для передачи). После успешного приема пакета приемник данных посылает *пакет подтверждения* (*Handshake Packet*).

Планирование транзакций обеспечивает управление поточными каналами. На аппаратном уровне использование от-

каза от транзакции (*NAck*) при недопустимой интенсивности передачи предохраняет буферы от переполнения сверху и снизу. Маркеры отвергнутых транзакций повторно передаются в свободное для шины время. Управление потоками позволяет гибко планировать обслуживание одновременных разнородных потоков данных.

Устойчивость к ошибкам обеспечивают следующие свойства USB:

- Высокое качество сигналов, достигаемое благодаря дифференциальным приемникам/передатчикам и экранированным кабелям.
- Защита полей управления и данных CRC-кодами.
- Обнаружение подключения и отключения устройств и конфигурирование ресурсов на системном уровне.
- Самовосстановление протокола с тайм-аутом при потере пакетов.
- Управление потоком для обеспечения изохронности и управления аппаратными буферами.
- Независимость функций от неудачных обменов с другими функциями.

Для обнаружения ошибок передачи каждый пакет имеет контрольные поля CRC-кодов, позволяющие обнаруживать все одиночные и двойные битовые ошибки. Аппаратные средства обнаруживают ошибки передачи, а контроллер автоматически производит трехкратную попытку передачи. Если повторы безуспешны, сообщение об ошибке передается клиентскому ПО.

Форматы пакетов

Байты передаются по шине последовательно, начиная с младшего бита. Все посылки организованы в пакеты. Каждый пакет начинается с поля синхронизации *Sync*, которое представляется последовательностью состояний *KJKJKJKK* (кодированную по NRZI), следующую после состояния *Idle*. Последние два бита (*KK*) являются маркером начала пакета *SOP*, используемым для идентификации первого бита идентификатора пакета *PID*. Идентификатор пакета является 4-бит-

ным полем *PID[3:0]*, идентифицирующим тип пакета (табл. 7.2), за которым в качестве контрольных следуют те же 4 бита, но инвертированные.

Таблица 7.2. Типы пакетов и их идентификаторы PID

Тип PID	Имя PID	PID[3:0]	Содержимое и назначение
Token	*OUT*	0001	Адрес функции и номер конечной точки – маркер транзакции функции
Token	*IN*	1001	Адрес функции и номер конечной точки – маркер транзакции хоста
Token	*SOF*	0101	Маркер начала кадра
Token	*SETUP*	1101	Адрес функции и номер конечной точки – маркер транзакции с управляющей точкой
Data	*Data0* *Data1*	0011 1011	Пакеты данных с четным и нечетным PID чередуются для точной идентификации подтверждений
Handshake	*Ack*	0010	Подтверждение безошибочного приема пакета
Handshake	*NAK*	1010	Приемник не сумел принять или передатчик не сумел передать данные. Может использоваться для управления потоком данных (неготовность). В транзакциях прерываний является признаком отсутствия необслуженных прерываний
Handshake	*STALL*	1110	Конечная точка требует вмешательства хоста
Special	*PRE*	1100	Преамбула передачи на низкой скорости

В пакетах-маркерах *IN*, *SETUP* и *OUT* следующими являются *адресные поля*: 7-битный адрес функции и 4-битный адрес конечной точки. Они позволяют адресовать до 127 функций USB (нулевой адрес используется для конфигурирования) и по 16 конечных точек в каждой функции.

В пакете *SOF* имеется 11-битное *поле номера кадра* (*Frame Number Field*), последовательно (циклически) увеличиваемое для очередного кадра.

Поле данных может иметь размер от 0 до 1023 целых байт. Размер поля зависит от типа передачи и согласуется при установлении канала.

Поле CRC-кода присутствует во всех маркерах и пакетах данных, оно защищает все поля пакета, исключая *PID*. CRC для маркеров (5 бит) и данных (11 бит) подсчитываются по разным формулам.

Каждая транзакция инициируется хост-контроллером посылкой маркера и завершается пакетом квитирования. Последовательность пакетов в транзакциях иллюстрирует рис. 7.7.

Хост-контроллер организует обмены с устройствами согласно своему плану распределения ресурсов. Контроллер циклически (с периодом 1 мс) формирует *кадры* (*Frames*), в которые укладываются все запланированные транзакции. Каждый кадр начинается с посылки маркера *SOF* (*Start Of Frame*), который является синхронизирующим сигналом для всех устройств, включая хабы. В конце каждого кадра выделяется интервал времени *EOF* (*End Of Frame*), на время которого хабы запрещают передачу по направлению к контроллеру. Каждый кадр имеет свой номер. Хост-контроллер оперирует 32-битным счетчиком, но в маркере *SOF* передает только младшие 11 бит. Номер кадра увеличивается (циклически) во время *EOF*. Хост планирует загрузку кадров так, чтобы в них всегда находилось место для транзакций управления и прерывания. Свободное время кадров может заполняться сплошными передачами (*Bulk Transfers*).

Рис. 7.7. Последовательности пакетов

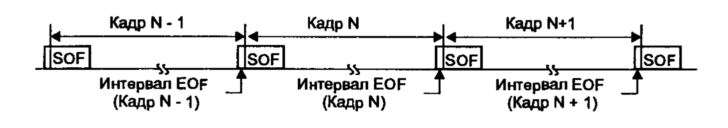


Рис. 7.8. Поток кадров USB

Для *изохронной передачи* важна синхронизация устройств и контроллера. Есть три варианта:

- синхронизация внутреннего генератора устройства с маркерами *SOF*;
- подстройка частоты кадров под частоту устройства;
- согласование скорости передачи (приема) устройства с частотой кадров.

Подстройка частоты кадров контроллера возможна, естественно, под частоту внутренней синхронизации только одного устройства. Подстройка осуществляется через механизм обратной связи, который позволяет изменять период кадра в пределах ±1 битового интервала.

7.1.2. Системное конфигурирование

USB поддерживает динамическое подключение и отключение устройств. Нумерация устройств шины является постоянным процессом, отслеживающим изменения физической топологии.

Все устройства подключаются через порты хабов. Хабы определяют подключение и отключение устройств к своим портам и сообщают состояние портов при запросе от контроллера. Хост разрешает работу порта и адресуется к устройству через канал управления, используя *нулевой адрес* — *USB Default Address*. При начальном подключении или после сброса все устройства адресуются именно так.

Хост определяет, является новое подключенное устройство хабом или функцией, и назначает ему *уникальный адрес* USB. Хост создает канал управления (*Control Pipe*) с этим устройством, используя назначенный адрес и нулевой номер точки назначения.

Если новое устройство является хабом, хост определяет подключенные к нему устройства, назначает им адреса и уста-

навливает каналы. Если новое устройство является функцией, уведомление о подключении передается диспетчером USB заинтересованному ПО.

Когда устройство отключается, хаб автоматически запрещает соответствующий порт и сообщает об отключении контроллеру, который удаляет сведения о данном устройстве из всех структур данных. Если отключается хаб, процесс удаления выполняется для всех подключенных к нему устройств. Если отключается функция, уведомление посылается заинтересованному ПО.

Нумерация устройств, подключенных к шине (*Bus Enumeration*), осуществляется динамически по мере их подключения (или включения их питания) без какого-либо вмешательства пользователя или клиентского ПО. Процедура нумерации выполняется следующим образом:

1. Хаб, к которому подключилось устройство, информирует хост о смене состояния своего порта ответом на опрос состояния. С этого момента устройство переходит в состояние *Attached* (подключено), а порт, к которому оно подключилось, в состояние *Disabled.*
2. Хост уточняет состояние порта.
3. Узнав порт, к которому подключилось новое устройство, хост дает команду сброса и разрешения порта.
4. Хаб формирует сигнал Reset для данного порта (10 мс) и переводит его в состояние *Enabled.* Подключенное устройство может потреблять от шины ток питания до 100 мА. Устройство переходит в состояние *Powered* (питание подано), все его регистры переводятся в исходное состояние, и оно отзывается на обращение по нулевому адресу.
5. Пока устройство не получит уникальный адрес, оно доступно по дежурному каналу, по которому хост-контроллер определяет максимально допустимый размер поля данных пакета.
6. Хост сообщает устройству его уникальный адрес, и оно переводится в состояние *Addressed* (адресовано).

7. Хост считывает конфигурацию устройства, включая заявленный потребляемый ток от шины. Считывание может затянуться на несколько кадров.
8. Исходя из полученной информации, хост конфигурирует все имеющиеся конечные точки данного устройства, которое переводится в состояние *Configured* (сконфигурировано). Теперь хаб позволяет устройству потреблять от шины полный ток, заявленный в конфигурации. Устройство готово.

Когда устройство отключается от шины, хаб уведомляет об этом хост и работа порта запрещается, а хост обновляет свою текущую топологическую информацию.

7.1.3. Устройства USB – функции и хабы

Возможности шины USB позволяют использовать ее для подключения разнообразных устройств. Не касаясь «полезных» свойств ПУ, остановимся на их интерфейсной части, связанной с шиной USB. Все устройства должны поддерживать набор общих операций, перечисленных ниже.

Динамическое подключение и отключение. Эти события отслеживаются хабом, который сообщает о них хост-контроллеру и выполняет сброс подключенного устройства. Устройство после сигнала сброса должно отзываться на нулевой адрес, при этом оно не сконфигурировано и не приостановлено. После назначения адреса, за которое отвечает хост-контроллер, устройство должно отзываться только на свой уникальный адрес.

Конфигурирование устройств, выполняемое хостом, является необходимым для их использования. Для конфигурирования обычно используется информация, считанная из самого устройства. Устройство может иметь множество интерфейсов, каждому из которых соответствует собственная конечная точка, представляющая хосту функцию устройства. Интерфейс в конфигурации может иметь альтернативные наборы характеристик; смена наборов поддерживается протоколом. Для поддержки адаптивных драйверов дескрипторы устройств и интерфейсов имеют поля класса, подкласса и протокола.

Передача данных возможна посредством одного из четырех типов передач (см. выше). Для конечных точек, допускающих разные типы передач, после конфигурирования доступен только один из них.

Управление энергопотреблением является весьма развитой функцией USB. Для устройств, питающихся от шины, мощность ограничена. Любое устройство при подключении не должно потреблять от шины ток, превышающий 100 мА. Рабочий ток (не более 500 мА) заявляется в конфигурации, и если хаб не сможет обеспечить устройству заявленный ток, оно не конфигурируется и, следовательно, не может быть использовано.

Устройство USB должно поддерживать *приостановку* (*Suspended Mode*), в котором его потребляемый ток не превышает 500 мкА. Устройство должно автоматически приостанавливаться при прекращении активности шины.

Возможность *удаленного пробуждения* (*Remote Wakeup*) позволяет приостановленному устройству подать сигнал хост-компьютеру, который тоже может находиться в приостановленном состоянии. Возможность удаленного пробуждения описывается в конфигурации устройства. При конфигурировании эта функция может быть запрещена.

Хаб в USB выполняет коммутацию сигналов и выдачу питающего напряжения, а также отслеживает состояние подключенных к нему устройств, уведомляя хост об изменениях. Хаб состоит из двух частей — контроллера (*Hub Controller*) и повторителя (*Hub Repeater*). *Повторитель* представляет собой управляемый ключ, соединяющий выходной порт со входным. Он имеет средства поддержки сброса и приостановки передачи сигналов. *Контроллер* содержит регистры для взаимодействия с хостом. Доступ к регистрам осуществляется по специфическим командам обращения к хабу. Команды позволяют конфигурировать хаб, управлять нисходящими портами и наблюдать их состояние.

Нисходящие (*Downstream*) *порты* хабов могут находиться в следующих состояниях:

- *Powered off* (питание отключено) — на порт не подается питание (возможно только для хабов, коммутирующих

питание). Выходные буферы переводятся в высокоимпедансное состояние, входные сигналы игнорируются.

- *Disconnected* (отсоединен) — порт не передает сигналы ни в одном направлении, но способен обнаружить подключение устройства (по отсутствию состояния *SE0* в течение 2,5 мкс). Тогда порт переходит в состояние *Disabled*, а по уровням входных сигналов (*Diff0* или *Diff1* в состоянии *Idle*) он определяет скорость подключенного устройства.
- *Disabled* (запрещен) — порт передает только сигнал сброса (по команде от контроллера), сигналы от порта (кроме обнаружения отключения) не воспринимаются. По обнаружении отключения (2,5 мкс состояния *SE0*) порт переходит в состояние *Disconnect*, а если отключение обнаружено «спящим» хабом, контроллеру будет послан сигнал *Resume*.
- *Enabled* (разрешен) — порт передает сигналы в обоих направлениях. По команде контроллера или по обнаружении ошибки кадра порт переходит в состояние *Disabled*, а по обнаружении отключения — в состояние *Disconnect*.
- *Suspended* (приостановлен) — порт передает сигнал перевода в состояние останова («спящий» режим). Если хаб находится в активном состоянии, сигналы через порт не пропускаются ни в одном направлении. Однако «спящий» хаб воспринимает сигналы смены состояния незапрещенных портов, подавая «пробуждающие» сигналы от активизировавшегося устройства даже через цепочку «спящих» хабов.

Состояние каждого порта идентифицируется контроллером хаба с помощью отдельных регистров. Имеется общий регистр, биты которого отражают факт изменения состояния каждого порта (фиксируемый во время *EOF*). Это позволяет хост-контроллеру быстро узнать состояние хаба, а в случае обнаружения изменений специальными транзакциями уточнить состояние.

7.1.4. Хост-контроллер

Хост-компьютер общается с устройствами через контроллер. Хост имеет следующие обязанности:

- обнаружение подключения и отсоединения устройств USB;
- манипулирование потоком управления между устройствами и хостом;
- управление потоками данных;
- сбор статистики;
- обеспечение энергосбережения подключенными ПУ.

Системное ПО контроллера управляет взаимодействием между устройствами и их ПО, функционирующим на хост-компьютере, для согласования:

- нумерации и конфигурации устройств;
- изохронных передач данных;
- асинхронных передач данных;
- управления энергопотреблением;
- информации об управлении устройствами и шиной.

По возможности ПО USB использует существующее системное ПО хост-компьютера — например, Advanced Power Management для управления энергопотреблением.

7.2. Шина IEEE 1394 — FireWire

Стандарт для высокопроизводительной последовательной шины (High Performance Serial Bus), получивший официальное название IEEE 1394, был принят в 1995 году. Целью являлось создание шины, не уступающей современным стандартным параллельным шинам, при существенном удешевлении и повышении удобства подключения (за счет перехода на последовательный интерфейс). Стандарт основан на шине *FireWire*, используемой Apple Computer в качестве дешевой альтернативы SCSI в компьютерах Macintosh и PowerMac. Название FireWire («огненный провод») теперь применяется и к реализациям IEEE 1394, оно сосуществует с кратким обозначением *1394*.

Преимущества FireWire перед другими последовательными шинами:

- Многофункциональность: шина обеспечивает цифровую связь до 63 устройств без применения дополнительной аппаратуры (хабов). Устройства — цифровые камкодеры, сканеры, принтеры, камеры для видеоконференций, дисковые накопители — могут обмениваться данными не только с PC, но и между собой. FireWire по инициативе VESA позиционируется и для «домашних сетей».
- Высокая скорость обмена и изохронные передачи позволяют даже на начальном уровне (100 Мбит/с) передавать одновременно два канала видео (30 кадров в секунду) широковещательного качества и стереоаудиосигнал с качеством CD.
- Низкая цена компонентов и кабеля.
- Легкость установки и использования. FireWire расширяет систему PnP. Устройства автоматически распознаются и конфигурируются при включении/отключении. Питание от шины (ток до 1,5 А) позволяет ПУ общаться с системой даже при отключении их питания. Управлять шиной и другими устройствами могут не только PC, но и другие «интеллектуальные» устройства, например VCR.

7.2.1. Структура и взаимодействие устройств шины

Стандарт 1394 определяет две категории шин: кабельные шины и кросс-шины (*Backplane*). Под *кросс-шинами* обычно подразумеваются параллельные интерфейсы, объединяющие внутренние подсистемы устройства, подключенного к кабелю 1394.

В отличие от USB, управляемой одним хост-контроллером, стандарт 1394 допускает соединение равноправных устройств в сеть. Сеть может состоять из множества шин, соединенных мостами. В пределах одной шины устройства объединяются соединительными кабелями без применения дополнительных устройств. *Мосты* представляют собой специальные интеллектуальные устройства. Интерфейсная карта шины FireWire для PC представляет собой мост PCI — 1394. Мостами являются также соединения кабельной шины 1394 с кросс-шинами устройств. 16-битная адресация узлов сети допус-

кает до 63 устройств в каждой шине, адресуемых 6-битным полем идентификатора узла. 10-битное поле идентификатора шины допускает использование в системе до 1023 мостов, соединяющих шины разного типа.

Кабельная шина представляет собой сеть, состоящую из узлов и кабельных мостов. Гибкая топология позволяет строить сети, сочетающие древовидную и цепочечную архитектуры (рис. 7.9). Каждый узел обычно имеет три равноправных соединительных разъема. Допускается множество вариантов подключения устройств со следующими ограничениями:

- между любой парой узлов может быть не более 16 кабельных сегментов;
- длина сегмента стандартного кабеля не должна превышать 4,5 м;
- суммарная длина кабеля не должна превышать 72 м (применение более качественного кабеля позволяет ослабить это ограничение).

Некоторые устройства могут иметь только один разъем, что ограничивает возможные варианты их местоположения. Стандарт допускает до 27 разъемов на одном устройстве.

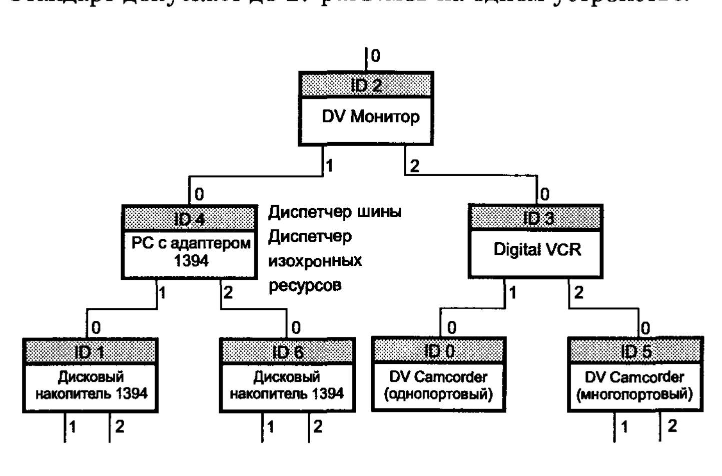


Рис. 7.9. Соединение устройств на шине FireWire

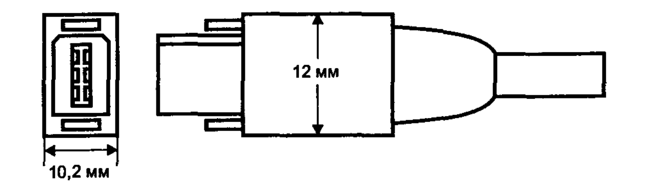


Рис. 7.10. Разъем FireWire

Стандарт предусматривает связь узлов с помощью 6-проводного кабеля, заключенного в общий экран. Две витые пары используются для передачи сигналов (раздельные для приемника и передатчика), два провода задействованы для питания устройств (8–40 В, до 1,5 А). Для гальванической развязки интерфейса используются трансформаторы (напряжение изоляции развязки до 500 В) или конденсаторы (в дешевых устройствах с напряжением развязки до 60 В относительно общего провода). Представление о разъемах дает рис. 7.10. Некоторые устройства (камкодеры Sony DCR-VX700 и DCR-VX1000, а также DHR-1000 DVCR) имеют только один 4-контактный разъем меньшего размера, у которого реализованы только сигнальные цепи. Эти устройства подключаются к шине через специальный переходной кабель только как оконечные (хотя возможно применение специальных адаптеров-разветвителей).

Стандарт 1394 определяет три возможные частоты передачи сигналов по кабелям: 98,304, 196,608 и 393,216 Мбит/с, которые округляют до 100, 200 и 400 Мбит/с. Частоты в стандарте обозначаются как *S100*, *S200* и *S400* соответственно. Бытовые устройства обычно поддерживают *S100*, большинство адаптеров допускают *S200*. К одной шине могут подключаться устройства, рассчитанные на разные скорости. Обмен будет происходить на минимальной для всех активных узлов скорости. Однако, если хост-контроллер реализует карту топологии и скоростей (*Topology_Map* и *Speed_Map*), возможно использование нескольких частот в одной шине, в соответствии с возможностями конкретной пары, участвующей в обмене.

Система допускает динамическое (горячее) подключение и отключение устройств. Идентификаторы подключаемым

устройствам назначаются автоматически, без участия пользователя. Изменения топологии (состава подключенных устройств) автоматически отслеживаются шиной и передаются управляющему ПО.

Протокол IEEE 1394

Протокол 1394 реализуется на трех уровнях (рис. 7.11).

- *Уровень транзакций* (*Transaction Layer*) преобразует пакеты в данные, предоставляемые приложениям, и наоборот. Он реализует протокол запросов-ответов, соответствующий стандарту ISO/IEC 13213:1994 (ANSI/IEEE 1212, редакции 1994 г.), архитектуры регистров управления и состояния CSR (Control and Status Register) для микрокомпьютерных шин (чтение, запись, блокировка). Это облегчает связь шины 1394 со стандартными параллельными шинами.
- *Уровень связи* (*Link Layer*) из данных физического уровня формирует пакеты и выполняет обратные преобразования. Он обеспечивает обмен узлов датаграммами с подтверждениями. Уровень отвечает за передачу пакетов и управление изохронными передачами.
- *Физический уровень* (*Physical Layer*) вырабатывает и принимает сигналы шины. Он обеспечивает инициализацию и арбитраж, предполагая, что в любой момент времени работает только один передатчик. Уровень передает потоки данных и уровни сигналов последовательной шины вышестоящему уровню. Между этими уровнями возможна гальваническая развязка, при которой микросхемы физического уровня питаются от шины. Гальваническая развязка необходима для предотвращения паразитных контуров общего провода, которые могут появиться через провода защитного заземления блоков питания.

Аппаратная часть FireWire обычно состоит из двух специализированных микросхем — трансиверов физического уровня *PHY Transceiver* и моста связи с шиной *LINK Chip*. Связь между ними возможна, например, по интерфейсу IBM-Apple LINK-PHY. Микросхемы уровня связи выполняют все функции своего уровня и часть функций уровня

транзакций, остальная часть уровня транзакций выполняется программно.

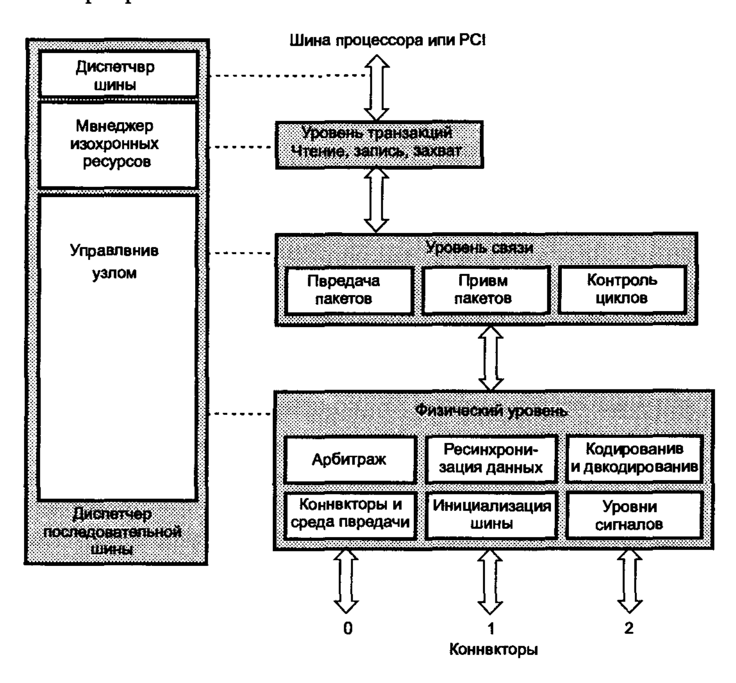


Рис. 7.11. Трехуровневая структура FireWire

Управление шиной

Протокол 1394 имеет гибкий механизм управления связью между различными устройствами. Для этого не обязательно присутствие на шине PC или иного контроллера шины. Управление включает три сервиса:

- *Мастер циклов*, посылающий широковещательные пакеты начала циклов (требуемые для изохронных обменов).
- *Диспетчер изохронных ресурсов*, если какой-либо узел поддерживает изохронный обмен (для цифрового видео и аудио).
- *Необязательный контроллер шины* (*Bus Master*) — им может являться PC или редактирующий DVCR.

По сбросу производится определение структуры шины, каждому узлу назначаются физические адреса и производится арбитраж мастера циклов, диспетчера изохронных ресурсов и контроллера шины. Через секунду после сброса все ресурсы становятся доступными для последующего использования.

Принципиальным преимуществом шины является отсутствие необходимости в контроллере. Любое передающее устройство может получить полосу изохронного трафика и начинать передачу по сигналу автономного или дистанционного управления — приемник «услышит» эту информацию. При наличии контроллера (PC) соответствующее ПО может управлять работой устройств, реализуя, например, цифровую студию нелинейного видеомонтажа.

Изохронная транспортировка данных

Изохронная транспортировка шины 1394 обеспечивает гарантированную пропускную способность и ограниченную задержку при высокоскоростной передаче по множеству каналов. Диспетчер изохронных ресурсов содержит регистр *BANDWIDTH_AVAILABLE*, который определяет доступность оставшейся части полосы пропускания для узлов с изохронной передачей. По сбросу вновь появившийся узел с изохронной передачей запрашивает выделение полосы. Для цифрового видео, например, требуется полоса 30 Мбит/с (25 Мбит/с на видеоданные и 3–4 Мбит/с на аудио, синхронизацию и заголовки пакетов). Полоса измеряется в специальных единицах распределения, число которых в 125-миллисекундном цикле составляет 6144. Единица занимает около 20 нс, что соответствует времени, требуемому для передачи одного *квадлета* (Quadlet) на частоте 1600 Мбит/с. *Квадлет* (32-битное слово) является единицей передачи данных по шине. 25 мс цикла резервируется под асинхронный трафик, поэтому начальное значение регистра после сброса составляет 4915 единиц. В *S100* устройства цифрового видео запрашивают около 1800 единиц, в *S200* — около 900. Если соответствующая полоса недоступна, запрашивающее ее устройство будет периодически повторять запрос.

Диспетчер изохронных ресурсов каждому изохронному узлу назначает номер канала (0–63) из числа доступных (регистр

CHANNELS_AVAILABLE). Он является идентификатором изохронного пакета. Когда изохронный обмен становится ненужным узлу, он должен освободить свою полосу и номер канала. Обмен управляющей информацией производится по асинхронному каналу.

7.2.2. Синонимы и дополнения стандарта IEEE 1394

Шина IEEE 1394 имеет множество псевдонимов:

- IEEE 1394-1995 Standard for a High Performance Serial Bus — полное название документа, описывающего стандарт, действующий в настоящее время.
- FireWire — торговая марка реализации IEEE-1394 фирмой Apple Computer, Inc.
- P1394 — название предварительной версии IEEE-1394 (до принятия в декабре 1995 г.).
- DigitalLink — торговая марка Sony Corporation, используемая применительно к реализации IEEE-1394 в цифровых камерах.
- MultiMedia Connection — имя, используемое в логотипе 1394 High Performance Serial Bus Trade Association (1394TA).

Поскольку фирма Apple разрабатывала концепцию FireWire еще с 1986 года, имя FireWire является самым распространенным синонимом IEEE 1394.

Кроме основного стандарта IEEE1394-1995, имеется несколько его модификаций:

- 1394a рассматривается как чистовой документ, заполняющий некоторые пробелы исходного стандарта и имеющий небольшие изменения (например, ускоренную операцию сброса на шине). Продуктам 1394a обеспечена обратная совместимость с устройствами, выпущенными до принятия основного стандарта. Версия вводилась для повышения скорости до 800 Мбит/с и выше, высокоскоростные версии входят и в 1394b.
- 1394.1 определяет 4-проводный соединитель и устанавливает стандарт на шинные мосты.

- 1394.2 предполагается как стандарт на соединение кластера станций со скоростью обмена 1 Гбит/с и выше, *несовместимый* с 1394. Этот стандарт проистекает из IEEE 1596 SCI (Scalable Coherent Interface — масштабируемый когерентный интерфейс) для суперкомпьютеров и иногда называется *Serial Express* или *SCILite*. Сигнальный интерфейс 1394.2 похож на FCAL и допускает *кольцевую топологию*, запрещаемую стандартом 1394.

7.2.3. Сравнение FireWire и USB

Последовательные интерфейсы FireWire и USB, имея общие черты, являются существенно различными технологиями. Обе шины обеспечивают простое подключение большого числа ПУ (127 для USB и 63 для FireWire), допуская коммутации и включение/выключение устройств при работающей системе. Топология обеих шин достаточно близка. Хабы USB входят в состав ЦУ; для пользователя их присутствие незаметно. Обе шины имеют линии питания устройств, но допустимая мощность для FireWire значительно выше. Обе шины поддерживают систему PnP (автоматическое конфигурирование при включении/выключении) и снимают проблему дефицита адресов, каналов DMA и прерываний. Различаются пропускная способность и управление шиной.

USB ориентирована на ПУ, подключаемые к PC. Ее изохронные передачи позволяют передавать только цифровые аудиосигналы. Все передачи управляются централизованно, и PC является необходимым управляющим узлом, находящимся в корне древовидной структуры шины. Соединение нескольких PC этой шиной не предусматривается.

FireWire ориентирована на интенсивный обмен между любыми подключенными к ней устройствами. Изохронный трафик позволяет передавать «живое» видео. Шина не требует централизованного управления со стороны PC. Возможно использование шины для объединения нескольких PC и ПУ в локальную сеть.

Новые устройства цифрового видео и аудио имеют встроенные адаптеры 1394. Подключение к шине FireWire традиционных аналоговых и цифровых устройств (плейеров, камер,

мониторов) возможно через адаптеры-преобразователи интерфейсов и сигналов. Стандартные однотипные кабели и разъемы FireWire заменяют множество разнородных соединений устройств бытовой электроники с PC. Разнотипные цифровые сигналы мультиплексируются в одну шину. В отличие от сетей Ethernet, высокоскоростные передачи потоков данных по FireWire в реальном времени не требуют дополнительных протоколов. Кроме того, имеются средства арбитража, гарантирующие доступ к шине за заданное время. Применение мостов в сетях FireWire позволяет изолировать трафик групп узлов друг от друга.

7.3. Шина ACCESS.Bus и интерфейс I²C

Последовательная шина *ACCESS.Bus* (Accessory Bus), разработанная фирмой DEC, является шиной взаимодействия компьютера с его аксессуарами — например, монитором (канал VESA DDC), интеллектуальными источниками питания (Smart Battery) и т. п. Шина позволяет по двум сигнальным и двум питающим (12 В, 500 мА) проводам подключить до 14 устройств ввода/вывода, длина шины может достигать 8 м. Аппаратной основой является интерфейс I²C, характеризуемый простотой реализации, но, даже по сравнению с USB, низкой производительностью. Над аппаратным протоколом I²C для шины ACCESS.Bus имеется базовый программный протокол, с которым взаимодействуют протоколы конкретных подключенных устройств. Протоколы обеспечивают подключение/отключение устройств без перезагрузки ОС. Назначение сигналов разъема ACCESS.Bus, предложенное VESA, приведено в табл. 7.3.

Таблица 7.3. Разъем ACCESS.Bus (VESA)

Контакт	Назначение
1	GND
2	Ключ
3	SDA
4	+5 В (питание устройств)
5	SCL

Интерфейс I²C, разработанный фирмой Philips, в PC появился недавно и используется как внутренняя вспомогательная шина системной платы для общения с энергонезависимой памятью идентификации установленных компонентов (модулей памяти DIMM). Шина отличается предельной простотой реализации — две сигнальные линии, с которыми работают программно. По прямому назначению эту шину применяет пока только BIOS при определении аппаратных средств, но использование перезаписываемой памяти конфигурирования открывает новые возможности для привязки ПО к конкретной системе (точнее, установленному модулю) и... для вирусов. Способ программного доступа к шине пока не стандартизован, но при желании его можно «вычислить», изучив документацию на чипсет.

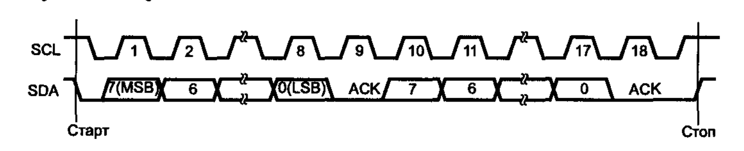


Рис. 7.12. Протокол передачи данных I²C

Последовательный интерфейс I²C обеспечивает двунаправленную передачу данных между парой устройств, используя два сигнала: данные SDA (Serial Data) и синхронизацию SCL (Serial Clock). В обмене участвуют два устройства — *ведущее* (*Master*) и *ведомое* (*Slave*). Каждое из них может выступать в роли *передатчика*, помещающего на линию SDA информационные биты, или *приемника*. Протокол обмена иллюстрирует рис. 7.12. Синхронизацию задает ведущее устройство — контроллер. Линия данных — двунаправленная с выходом типа «открытый коллектор» — управляется обоими устройствами поочередно. Частота обмена (не обязательно постоянная) ограничена сверху величиной 100 кГц для стандартного режима и 400 кГц для скоростного, что позволяет организовать программно-управляемую реализацию контроллера интерфейса.

Начало любой операции — условие *Start* — инициируется переводом сигнала SDA из высокого в низкий при высоком уровне SCL. Завершается операция переводом сигнала SDA

из низкого уровня в высокий при высоком уровне SCL — условие *Stop*. При передаче данных состояние линии SDA может изменяться только при низком уровне SCL, биты данных стробируются положительным перепадом SCL. Каждая посылка состоит из 8 бит данных, формируемых передатчиком (старший бит — MSB — передается первым), после чего передатчик на один такт освобождает линию данных для получения подтверждения. Приемник во время девятого такта формирует нулевой *бит подтверждения Ack*. После передачи бита подтверждения приемник может задержать следующую посылку, удерживая линию SCL на низком уровне. Приемник также может замедлить передачу по шине на уровне приема каждого бита, удерживая SCL на низком уровне после его спада, сформированного передатчиком.

Каждое ведомое устройство имеет свой адрес, разрядность которого по умолчанию составляет 7 бит. Адрес *A[6:0]* передается ведущим устройством в битах [7:1] первого байта, бит 0 содержит признак операции *RW* (*RW*=1 — чтение, *RW*=0 — запись). 7-битный адрес содержит две части: старшие 4 бита *A[6:3]* несут информацию о типе устройства (например, для EEPROM — 1010), а младшие 3 бита *A[0:2]* определяют номер устройства данного типа. Многие микросхемы с интерфейсом I^2C имеют три адресных входа, коммутацией которых на логические уровни 1 и 0 задается требуемый адрес. Некоторые значения полного адреса зарезервированы (табл. 7.4).

Общий вызов позволяет включившемуся устройству заявить о себе широковещательным способом. Байт *Start* предназначен для привлечения внимания процессора к интерфейсу, если в устройстве он организован программным (не аппаратным) способом. До получения этого байта микроконтроллер устройства не опрашивает состояние и не следит за сигналами интерфейса. При использовании 10-битной адресации биты [2:1] содержат старшую часть адреса, а младшие 8 бит будут переданы в следующем байте, если признак *RW*=0.

Адрес ведомого устройства и тип обращения задается контроллером при инициировании обмена. Обмен с памятью иллюстрирует рис. 7.13. Здесь SA[0:2] — адрес устройства, DA[0:7] — адрес данных, D[0:7] — данные, W — признак записи (0), R — признак чтения (1).

Таблица 7.4. Специальные адреса I²C

Биты [7:1]	Бит 0 (RW)	Назначение
0000 000	0	*General call address* — адрес общего вызова
0000 000	1	*Start* — начало активного обмена
0000 001	X	Адрес устройства шины CBUS (для совместимости)
0000 010	X	Адрес для устройств иных шин
0000 011	X	Зарезервировано
0000 1XX	X	Зарезервировано
1111 1XX	X	Зарезервировано
1111 0XX	X	Признак 10-битной адресации



Рис. 7.13. Обмен с памятью по интерфейсу I²C: *а* — запись, *б* — чтение с текущего адреса, *в* — чтение с произвольного адреса

Выполнив условие *Start*, контроллер передает байт, содержащий адрес устройства и признак операции *RW*, и ожидает подтверждения. При *операции записи* следующей посылкой от контроллера будет 8-битный адрес записываемой ячейки, а за ней — байт данных (для микросхем объемом памяти более 256 байт адрес ячейки посылается двумя байтами). Получив подтверждения, контроллер завершает цикл условием *Stop*, а адресованное устройство может начать свой внутренний цикл записи, во время которого не реагирует на сигналы интерфейса. Контроллер проверяет готовность устройства посылкой команды записи (байт адреса устройства)

и анализом бита подтверждения, формируя затем условие *Stop*. Если устройство откликнулось битом подтверждения, значит, оно завершило внутренний цикл и готово к следующей операции.

Операция считывания инициируется так же, как и запись, но с признаком RW=1. Возможно чтение по заданному адресу, по текущему адресу или последовательное. Текущий адрес хранится во внутреннем счетчике ведомого устройства, он содержит увеличенный на единицу адрес ячейки, участвовавшей в последней операции.

Получив команду чтения, устройство дает бит подтверждения и посылает байт данных, соответствующий текущему адресу. Контроллер может ответить подтверждением, тогда устройство пошлет следующий байт (последовательное чтение). Если на принятый байт данных контроллер ответит условием *Stop*, операция чтения завершается (случай чтения по текущему адресу). Начальный адрес для считывания контроллер задает фиктивной операцией записи, в которой передается байт адреса устройства и байт адреса ячейки, а после подтверждения приема байта адреса снова формируется условие *Start* и передается адрес устройства, но уже с указанием на операцию чтения. Так реализуется считывание произвольной ячейки (или последовательности ячеек).

Интерфейс позволяет контроллеру с помощью пары сигналов обращаться к любому из 8 однотипных устройств, подключенных к данной шине и имеющих уникальный адрес (рис. 7.14). При необходимости увеличения количества устройств возможно подключение групп. При этом допустимо как использование общего сигнала SCL и раздельных сигналов SDA (двунаправленных), так и общего сигнала SDA и раздельных однонаправленных сигналов SCL. Для обращения к одной из нескольких микросхем (или устройств), не имеющих выводов для задания собственного адреса, также применяют разделение линий SCL (или SDA).

Протокол I^2C позволяет нескольким контроллерам использовать одну шину, определяя коллизии и выполняя арбитраж. Эти функции реализуются достаточно просто: если два передатчика пытаются установить на линии SDA различные

логические уровни сигналов, то «победит» тот, который установит низкий уровень. Передатчик следит за уровнями управляемых им сигналов и при обнаружении несоответствия (передает высокий уровень, а «видит» низкий) отказывается от дальнейшей передачи. Устройство может инициировать обмен только при пассивном состоянии сигналов. Коллизия может возникнуть лишь при одновременной попытке начала обмена — как только конфликт обнаружен, «проигравший» передатчик отключится, а «победивший» продолжит работу.

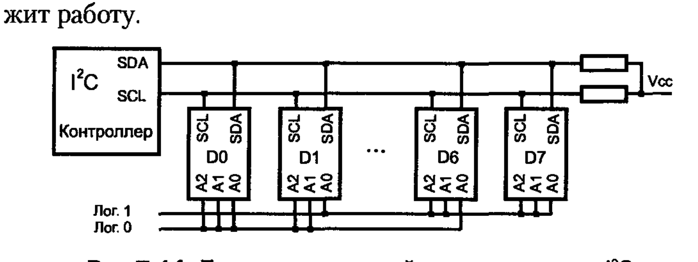


Рис.7.14. Подключение устройств к контроллеру I^2C

Приложение А. Системотехника IBM PC-совместимых компьютеров

Здесь рассмотрено взаимодействие программ с интерфейсными адаптерами. Приведены краткие сведения по архитектуре PC. Описаны организация пространств памяти и ввода/вывода, система прерываний и прямой доступ к памяти. Более подробные сведения можно найти в книге «Аппаратные средства IBM PC. Энциклопедия» («Питер», 1998).

А.1. Пространство памяти

Логическая структура памяти PC обусловлена системой адресации процессоров семейства x86. Процессоры 8086/88, применявшиеся в первых моделях IBM PC, имели адресное пространство 1 Мбайт (20 бит шины адреса). Начиная с процессора 80286 шина адреса была расширена до 24 бит, затем (386DX, 486, Pentium) до 32 и, наконец, до 36 бит (Pentium Pro, Pentium II). В реальном режиме процессора, используемом в DOS, формально доступен лишь 1 Мбайт памяти. Однако из-за ошибки эмуляции процессора 8086 в реальном режиме процессоры 80286 и выше имеют максимально доступный адрес 10FFEFh, что на (64К−16) байт больше. Область 100000h−10FFEFh называется *высокой памятью — High Memory Area* (HMA). В нее помещают часть ОС реального режима и небольшие резидентные программы. Для полной совместимости с процессором 8086/88 имеется вентиль линии A20 шины адреса — *GateA20*, который либо пропускает сигнал от процессора, либо принудительно обнуляет линию A20 системной шины адреса.

Основную часть адресного пространства занимает *оперативная память*. Объем установленной памяти определяется тестом POST при включении компьютера, начиная с младших адресов. Натолкнувшись на отсутствие памяти (ошибку), тест останавливается и сообщает системе объем памяти.

Память, непосредственно адресуемая процессором, распределяется следующим образом:

- 00000h–9FFFFh — *Conventional (Base) Memory*, 640 Кбайт — стандартная (базовая) память, доступная DOS и программам реального режима. В некоторых системах с видеоадаптером MDA верхняя граница сдвигается к AFFFFh (704 Кбайт). Иногда верхние 128 Кбайт стандартной памяти (область 80000h–9FFFFh) называют *Extended Conventional Memory*.
- A0000h–FFFFFh — *Upper Memory Area (UMA)*, 384 Кбайт — верхняя память, зарезервированная для системных нужд. В ней размещаются буферы интерфейсных адаптеров (например, видеопамять) и постоянная память (BIOS с расширениями).
- Память выше 100000h — *Extended Memory* — дополнительная (расширенная) память, непосредственно доступная только в защищенном режиме компьютеров 286 и выше.

Для компьютеров класса AT с 24-битной шиной адреса верхняя граница дополнительной памяти — FDFFFFh (максимальный размер 14,9 Мбайт). Область FE0000h–FFFFFFh содержит ПЗУ BIOS (ROM BIOS Area); обращение к ней эквивалентно обращению к ROM BIOS по адресам 0E0000h–0FFFFFh.

Для процессоров 386 и 32-битной шины адреса теоретическая верхняя граница — 4 Гбайт, а для Pentium Pro и Pentium II — 64 Гбайт (36-битная шина адреса). Обращение по адресам, превышающим границу установленной оперативной памяти (или максимально возможного объема), транслируется на шину PCI, которая имеет 32-битную адресацию. В компьютерах с 32-разрядной шиной адреса образ BIOS дополнительно проецируется в адреса FFFE0000h–FFFFFFFFh. Иногда используется проекция BIOS в область FE0000h–FFFFFFh, что не позволяет использовать более

16 Мбайт ОЗУ, поскольку система находит только непрерывную область расширенной памяти. Для использования специфических адаптеров ISA, имеющих буфер с адресами в 16-м мегабайте памяти, в BIOS Setup предусматривают опцию *Memory Hole At 15–16M*. Ее установка не позволяет использовать оперативную память свыше 16 Мбайт.

Для взаимодействия с интерфейсными адаптерами ISA интересна *верхняя память UMA*. Стандартное распределение верхней памяти выглядит следующим образом:

- A0000h–BFFFFh – *Video RAM*, 128 Кбайт – видеопамять (обычно используется не полностью).
- C0000h–DFFFFh – *Adapter ROM*, *Adapter RAM*, 128 Кбайт – зарезервировано для адаптеров, использующих собственные модули ROM BIOS и/или специальное ОЗУ, разделяемое с системной шиной.
- E0000h–EFFFFh – свободная область, 64 Кбайт, иногда занятая под System BIOS.
- F0000h–FFFFFh – *System BIOS*, 64 Кбайт – ROM на системной плате (в XT используется только FE000h–FFFFFh – 8 Кбайт).
- FD000h–FDFFFh – *ESCD* (Extended System Configuration Data) – область энергонезависимой памяти, используемая для конфигурирования устройств PnP. Эта область присутствует только при наличии PnP BIOS; ее положение и размер жестко не заданы.

В области UMA практически всегда присутствует графический адаптер. В зависимости от модели он занимает следующие области:

- *MDA RAM* – B0000h–B0FFFh
- *CGA RAM* – B8000h–BBFFFh
- *EGA ROM* – C0000h–C3FFFh/C7FFFh
- *VGA ROM* – C0000h–C7FFFh
- *EGA*, *VGA RAM* – A0000h–BFFFFh, в зависимости от видеорежима используются следующие области: *Graphics* – A0000h–AFFFFh; *Color Text* – B8000h–BFFFFh; *Mono Text* – B0000h–B7FFFh.

Распространенным потребителем UMA являются *расширения ROM BIOS*, расположенные на платах дисковых контроллеров, адаптеров SCSI, сетевых картах и т. д. Обычно они занимают область C8000h – CBFFFh/C9FFFh/C8FFFh (для дисковых контроллеров), но могут перемещаться при конфигурировании адаптеров.

А.2. Пространство ввода/вывода

Процессоры x86 имеют раздельную адресацию памяти и портов ввода/вывода. Инструкции ввода/вывода порождают шинные циклы обмена, в которых вырабатываются сигналы IORD# (чтение порта) и IOWR# (запись в порт) на шинах ввода/вывода. Эти сигналы отличают пространство ввода/вывода от пространства памяти, где соответствующие операции чтения и записи вырабатывают сигналы MEMRD# и MEMWR#. Для обращения к пространству ввода/вывода предназначены четыре основные инструкции процессора: *IN* (ввод в регистр), *OUT* (вывод из регистра), *INS* (ввод из порта в элемент строки памяти) и *OUTS* (вывод элемента из строки памяти). Последние две инструкции с префиксом повтора *REP* используются для быстрой пересылки блоков данных между портом и памятью. Разрядность слова, передаваемого за одну инструкцию ввода/вывода, может составлять 8, 16 или 32 бита. В зависимости от выравнивания адреса по границе слова и разрядности шины это слово может передаваться за один или несколько циклов шины.

Сигналы IORD# и IOWR# вырабатываются и в циклах DMA; в этом случае на шину адреса подается адрес памяти, к которой производится доступ, а не порта. Для блокировки срабатывания дешифратора адреса порта в цикле DMA в шине ISA имеется сигнал AEN. Селективное программное управление сигналом AENx для каждого слота шины EISA дает возможность изолировать пространства ввода/вывода портов, что используется для программного конфигурирования адресов адаптеров. В шине ISA разделение слотов не предусмотрено, что объясняет сложность реализации для нее системы PnP.

Для дешифрации адреса в оригинальном PC из 16 бит использовались только младшие 10 (A0–A9), что позволяет об-

ращаться к портам 0–3FFh. Старшие биты адреса, хотя и поступают на шину, устройствами игнорируются. Так, обращение по адресам 378h, 778h, B78h и F78h воспринимается устройствами одинаково. В адаптерах для шин MCA и PCI используются все 16 бит адреса. Современные системные платы полностью дешифруют адрес.

В реальном режиме процессора программе доступно все пространство адресов ввода/вывода. В защищенном режиме (в частности, в режиме виртуального процессора 86) можно программно ограничить пространство ввода/вывода, определив его максимальный размер. Внутри разрешенной области доступ может быть разрешен или запрещен для каждого конкретного адреса. Размер области и карта разрешенных портов (*IO Permission Bitmap*) задаются ОС в дескрипторе сегмента состояния задачи (TSS). При обращении по неразрешенному адресу вырабатывается исключение процессора, обработчик которого определяется ОС. Возможно снятие задачи-нарушителя. Не исключено, что по обращению к порту ОС выполнит некоторые действия, создав для программы иллюзию реальной операции ввода/вывода.

Несколько портов вывода могут иметь совпадающие адреса. Операции записи они будут отрабатывать нормально. Для операций считывания ситуация другая. Если несколько портов ввода имеют совпадающий адрес, при считывании они передают свои данные на шину. Если они находятся на одной физической шине, возникает конфликт. При прочих равных условиях для каждого бита шины «побеждает» порт, выводящий логический ноль. Так, если один порт «хочет» передать байт 0Fh, а другой – F0h, то процессор считает 00h. Если порты находятся на разных физических шинах (например, один в ISA, а другой – в PCI), конфликта не будет, поскольку шины отделены друг от друга буферами данных. Каждой шине назначается своя область ввода, и дешифратор, расположенный на системной плате, при чтении открывает соответствующие буферы, так что реально считываются данные только с одной шины. При записи в порты данные (и сигнал записи) обычно распространяются по всем шинам компьютера. Адреса 0h–0FFh отведены для устройств системной платы, чтение по этим адресам не распространя-

ется на шины расширения. Для современных плат со встроенной периферией и несколькими шинами (ISA, PCI) распределением адресов управляет BIOS через регистры конфигурирования чипсета.

Приведенные выше рассуждения справедливы и для пространства памяти, но, как правило, они актуальны только для портов ввода/вывода.

Карта распределения адресов ввода/вывода стандартных устройств приведена в табл. А.1. Подразумевается 10-битная дешифрация адреса.

Таблица А.1. Стандартная карта портов ввода/вывода

AT и PS/2	PC/XT	Назначение
000–00F	000–00F	Контроллер DMA #1 8237
010–01F		PS/2 – расширение DMA #1
020–021	020–021	Контроллер прерываний #1 – 8259A
040–05F	040–043	Таймер (PC/XT: 8253, AT: 8254)
060	060	Диагностический регистр POST (только запись)
	060–063	Системный интерфейс 8255
060, 064		Контроллер клавиатуры AT 8042
061		Источники NMI и управление звуком
070–07F		Память CMOS и маска NMI
080		Диагностический регистр
080–08F	080–083	Регистры страниц DMA
090–097		PS/2: микроканал, арбитр и т. д.
	0A0	Маска NMI
0A0–0BF		Контроллер прерываний #2 – 8259A
0C0–0DF		Контроллер DMA #2 8237A–5
0F0–0FF		Сопроцессор 80287
100–1EF		Управление микроканалом PS/2
170–177		Контроллер НЖМД #2 (IDE#2)
1F0–1F7		Контроллер НЖМД #1 (IDE#1)

Таблица А.1 (продолжение)

AT и PS/2	PC/XT	Назначение
200–207	200–20F	Игровой адаптер
	210–217	Блок расширений
238–23F		COM4
278–27F	278–27F	Параллельный порт LPT2 (LPT3 при наличии MDA)
	2A2–2A3	Часы MSM48321RS
2C0–2DF	2C0–2DF	EGA #2
2E0–2E7		COM4
2E8–2EF		COM4
2F8–2FF	2F8–2FF	COM2
300–31F		Плата-прототип
	320–32F	Жесткий диск XT
338–33F		COM3
370–377		Контроллер НГМД #2
376–377		Порты команд IDE#2
378–37F	378–37F	Параллельный порт LPT1 (LPT2 при наличии MDA)
380–38F	380–38F	Синхронный адаптер SDLC/BSC #2
3A0–3AF	3A0–3A9	Синхронный адаптер BSC #1
3B0–3BB	3B0–3BB	Монохромный адаптер (MDA)
3B4–3C9		Видеосистема PS/2
3BC–3BF	3BC–3BF	Параллельный порт LPT1 платы MDA
3C0–3CF	3C0–3CF	EGA #1
3C0–3DF	3C0–3DF	VGA
3D0–3DF	3D0–3DF	CGA/EGA
3E0–3E7		COM3
3E8–3EF		COM3
3F0–3F7	3F0–3F7	Контроллер НГМД #1
3F6–3F7		Порты команд IDE#1
3F8–3FF	3F8–3FF	COM1

А.3. Аппаратные прерывания

Аппаратные прерывания обеспечивают реакцию процессора на события, происходящие асинхронно по отношению к исполняемому программному коду. Процессоры x86 поддерживают таблицу, содержащую определения до 256 процедур обслуживания прерываний. *Немаскируемые прерывания* NMI обрабатываются процессором независимо от состояния флага разрешения прерывания *IF*. К ним относятся прерывания, приходящие по линии NMI, а для процессоров, поддерживающих режим системного управления, еще и по линии SMI#.

Сигнал на линию NMI приходит от схем контроля паритета памяти, от линии IOCHK шины ISA или SERR# шины PCI. В машинах класса AT сигнал NMI блокируется до входа процессора установкой бита 7 порта 070h в 1, отдельные источники — битами 2, 3 порта 061h. Идентифицировать источник NMI позволяют биты 6, 7 регистра 061h.

В XT NMI вызывается и математическим сопроцессором при возникновении исключения. Запретить NMI позволяет обнуление бита 7 порта 0A0h; отдельные источники блокируются битами 4, 5 регистра 061h; биты 6, 7 регистра 062h идентифицируют источник.

Обработка *маскируемых прерываний* может запрещаться инструкцией *DI* и разрешаться — *EI* (или другим способом воздействия на флаг процессора *IF*). Эти прерывания обслуживаются контроллером, программно-совместимым с 8259A. Он имеет 8 входов запросов прерываний IRQx от внешних источников. При обработке запроса контроллер передает по шине данных 8-битный *вектор прерывания*, соответствующий номеру запроса. Этот вектор является индексом, по которому ссылка на процедуру обработки прерывания хранится в таблице прерываний.

В машинах класса AT используются два контроллера прерываний. Ведущий (первичный) контроллер 8259A#1 обслуживает запросы 0, 1, 3–7. К его входу 2 подключен ведомый (вторичный) контроллер 8259A#2, который обслуживает запросы 8–15. При этом используется вложенность приоритетов — запросы 8–15 со своим рядом убывающих приорите-

тов вклиниваются между запросами 1 и 3 ведущего контроллера, приоритеты запросов которого также убывают с ростом номера. В XT один контроллер 8259A обслуживал все 8 линий запросов.

На входы контроллеров прерываний поступают запросы от системных устройств и плат расширения. Эти линии обозначаются как IRQx и имеют общепринятое назначение (табл. A.2). Прерывания в табл. A.2 расположены в порядке убывания их приоритетов. Номера векторов, соответствующих линиям запросов контроллеров, система приоритетов и другие параметры могут задаваться программно при инициализации контроллеров.

Таблица А.2. Аппаратные прерывания

Имя (номер)	Вектор	Назначение
NMI	02h	Контроль канала, паритет (в XT – сопроцессор)
0	08h	Таймер (канал 0 8253/8254)
1	09h	Клавиатура
IRQ2	0Ah	XT – произвольно, AT – каскад IRQ8–IRQ15
8	70h	CMOS RTC – часы реального времени
IRQ9	71h	Произвольно
IRQ10	72h	Произвольно
IRQ11	73h	Произвольно
IRQ12	74h	PS/2-Mouse или произвольно
13	75h	Математический сопроцессор
IRQ14	76h	HDC – контроллер НЖМД
IRQ15	77h	Произвольно
IRQ3	0Bh	COM2, COM4
IRQ4	0Ch	COM1, COM3
IRQ5	0Dh	XT – HDC, AT – LPT2, Sound (произвольно)
IRQ6	0Eh	FDC – контроллер НГМД
IRQ7	0Fh	LPT1 – принтер

Назначение номеров прерываний — процесс двухсторонний: адаптер должен быть сконфигурирован на использование конкретной линии шины (джамперами или программно), а ПО, поддерживающее данный адаптер, должно быть проинформировано о номере используемого вектора. Поскольку прерывания являются дефицитным ресурсом, возникает желание разделить эти линии между несколькими устройствами. Тогда обработчик прерывания устройства, определив, что источник — не его, вызывал бы обработчик другого устройства, работающего с той же линией. Однако в шине ISA прерывание вырабатывается по *положительному перепаду сигнала* на линии запроса. Такой способ подачи сигнала имеет меньшую помехозащищенность, чем срабатывание по отрицательному перепаду, и отрезает путь к нормальному разделяемому использованию линий, для которого полностью пригоден способ подачи сигнала по низкому уровню. Поскольку традиционный контроллер позволяет задавать чувствительность — *уровень* (Level) или *перепад* (Edge) — только для всех входов одновременно, разделяемые прерывания на шине ISA неработоспособны. Тем не менее, некоторые чипсеты, реализующие контроллеры прерываний, допускают индивидуальное управление чувствительностью каждого входа. Тогда при соответствующих возможностях BIOS Setup и адаптеров разделяемые прерывания технически реализуемы.

При чувствительности к уровню сигнал запроса аппаратного прерывания IRQx должен удерживаться генерирующей схемой по крайней мере до цикла подтверждения прерывания процессором. В противном случае источник прерывания не будет правильно идентифицирован. Обычно адаптеры строят так, что сигнал запроса сбрасывается при обращении программы обслуживания прерывания к соответствующим регистрам адаптера.

Для запросов прерывания с шины PCI используются 4 линии, которые обозначают как INTR A, B, C, D. Возможно их разделяемое использование. Линии циклически сдвигаются в слотах (рис. А.1) и независимо коммутируются на доступные IRQx с помощью конфигурационных регистров чипсета. Линии IRQx, используемые шиной PCI, недоступны для шины ISA. «Дележку» линий между шинами, а также управление

чувствительностью отдельных линий выполняют настройки BIOS Setup, а также система PnP. В опциях настройки *ISA* или *Legacy* подразумевают использование линий IRQx традиционными адаптерами шины ISA (статическое распределение), а *PCI/PnP* — адаптерами шины PCI или адаптерами PnP для шины ISA (динамическое распределение).

Рис. А.1. Коммутация запросов прерывания шины PCI

А.4. Прямой доступ к памяти и прямое управление шиной

Прямой доступ к памяти — *DMA* (Direct Memory Access) — метод обмена ПУ с памятью без участия процессора. В обычном программном обмене (PIO) при пересылке блока данных, например, из порта в память (инструкция *REP INSB*), выполняются следующие действия:

- Процессор генерирует шинный цикл чтения порта, выставляя его адрес и формируя сигнал IORD#. Данные из порта считываются процессором во внутренний шинный буфер.
- Процессор генерирует шинный цикл записи в память, выставляя адрес ячейки и формируя сигнал MEMWR#. Данные из внутреннего шинного буфера записываются в память.
- Эти шаги автоматически повторяются с изменением адреса памяти. Количество повторов задается в регистре *CX*,

инкремент/декремент адреса — флагом *DF*. Во время передачи всего блока процессор занят.

В режиме прямого доступа к памяти процессор инициализирует контроллер прямого доступа к памяти — задает начальный адрес, счетчик и режим обмена, после чего освобождается. Обмен производит контроллер и выполняет его несколько иначе. Контроллер имеет несколько каналов. Для интерфейса ПУ каждый канал представляется парой сигналов: запрос обмена — DRQx и подтверждение обмена — DACKx#. В операциях по каналу DMA адрес порта не фигурирует, а используется только пара сигналов, соответствующая номеру канала. Цикл передачи блока в память будет выглядеть следующим образом (рис. А.2):

1. По сигналу DRQx контроллер запрашивает управление шиной и дожидается его предоставления процессором (и другими контроллерами шины).
2. Контроллер выставляет адрес ячейки памяти и формирует в одном цикле шины сигналы IORD#, DACKx# и MEMWR#. Сигнал DACKx# указывает на то, что операция выполняется для канала «х», а IORD# — на направление в канале (для пересылки из памяти в канал использовался бы сигнал IOWR#). Чтобы по сигналу IORD# не было ложного чтения (по IOWR# — ложной записи) порта, адрес которого совпадает с адресом памяти, присутствующим в цикле DMA, контроллер высоким уровнем сигнала AEN запрещает портам дешифрацию адреса. Байт, считанный из канала, в том же цикле шины записывается в ячейку памяти.
3. Контроллер модифицирует счетчик адреса и повторяет шаги 1–2 для каждого следующего сигнала DRQx, пока не будет исчерпан счетчик циклов. В последнем цикле обмена контроллер формирует общий сигнал окончания TC (Terminate Count), который может быть использован устройством для формирования сигнала аппаратного прерывания.

Обратная пересылка отличается только тем, что используются сигналы IOWR# и MEMRD#.

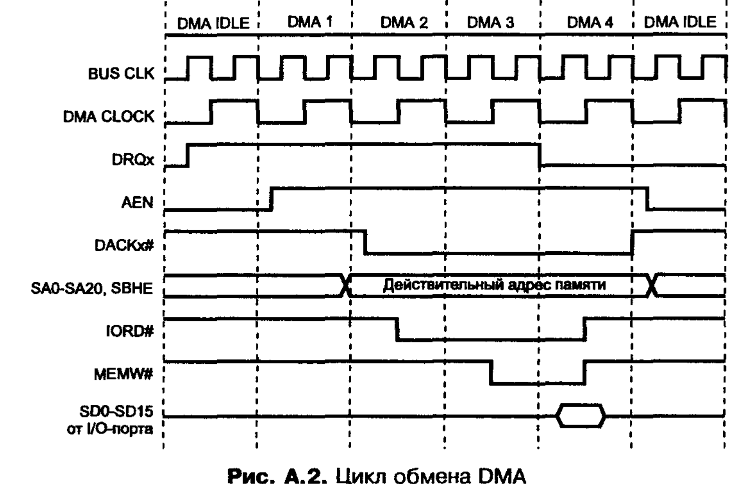


Рис. А.2. Цикл обмена DMA

Процессор при обмене занят только инициализацией контроллера, которая сводится к записи в его регистры нескольких байт. Затем обмен производят системная шина и контроллер. Если выбранный режим обмена не занимает всей пропускной способности шины, во время операций DMA процессор может продолжать работу.

Прямой доступ к памяти был использован еще в PC/XT, где применялась микросхема *четырехканального контроллера DMA* i8237A. Контроллер имеет 16-разрядные регистры адреса и счетчики, что позволяет программировать передачу блоков до 64 Кбайт. Для доступа к пространству размером в 1 Мбайт применили внешние 4-разрядные регистры страниц DMA, отдельные для каждого канала. В них хранятся биты адреса A[19:16], а битами A[15:0] управляет контроллер. При достижении регистром-счетчиком адреса значения FFFFh следующее его значение будет 0000h, а внешний регистр адреса страницы останется неизменным. Таким образом, если блок начинается не с границы страницы памяти размером в 64 Кбайт, возможно его «сворачивание в кольцо». Но если для процессоров 80x86 в реальном режиме «сворачиваемые» сегменты могут начинаться с адреса, кратного 10h, то при прямом доступе границы «сворачиваемого» сегмента кратны 10000h. Этот эффект обязательно необходимо учи-

тывать при программировании прямого доступа — блок, пересекающий границу, должен пересылаться за два сеанса DMA, между которыми канал (включая регистр страниц) должен быть повторно инициализирован.

Из четырех каналов DMA XT на шине ISA доступны только три (1, 2 и 3). Канал 0 используется для регенерации динамической памяти, адрес регенерируемой строки берется с линий адреса ISA. Каналы 1, 2 и 3 обеспечивают побайтную передачу данных и называются 8-битными каналами DMA.

В архитектуре AT подсистему DMA расширили, добавив второй контроллер 8237A. Его подключили к шине адреса со смещением на 1 бит. 16-битные регистры адреса способны управлять линиями адреса A[16:1], младший бит адреса A0 всегда нулевой. Таким образом, второй контроллер передает данные только по два байта, поэтому его каналы названы 16-битными. За один сеанс второй контроллер способен передать до 64 K 16-разрядных слов. Регистры страниц для всех каналов DMA у AT расширены до 8 бит, что делает доступной для любого канала область памяти размером 16 Мбайт (0–FFFFFFh). Эффект «сворачивания» сегментов в каналах прямого доступа сохраняется, только для каналов 5, 6 и 7 «кольца» имеют размер 64 K слов и границы, кратные 20000h. Микросхемы контроллеров 8237A включены каскадно, и канал 4 недоступен — через него подключается первый контроллер. Стандартное назначение каналов приведено в табл. A.3.

Кроме увеличения числа каналов в AT ввели *управление шиной ISA — Bus-Mastering* — со стороны адаптера. Оно опирается на контроллер DMA, выполняющий в данном случае функции арбитра шины. Для получения управления шиной адаптер посылает запрос по линии DRQx (только для каналов 5–7) и, получив подтверждение DACKx, устанавливает сигнал MASTER#. Теперь шиной ISA управляет он, но не имеет права занимать шину более чем на 15 мкс за сеанс. В противном случае нарушится регенерация памяти. Интеллектуальный контроллер может выполнять более эффективные процедуры обмена, чем стандартный DMA.

Управление шиной используют высокопроизводительные адаптеры SCSI и локальных сетей, а также интеллектуаль-

ные графические адаптеры. Однако архитектурой шины доступное им пространство ограничено 16 Мбайт.

Ограничения на доступную память для «чистого» режима DMA в новых компьютерах снимаются применением расширенных регистров страниц, но об этом должно «знать» ПО.

На шине EISA каналы DMA могут работать в 8-, 16- и 32-битном режиме и, используя все 32 разряда шины адреса, иметь доступ ко всей памяти компьютера. Каждый канал программируется на 1 из 4 типов цикла передачи:

- Compatible — полностью совместим с ISA.
- Type A — сокращенный на 25% цикл: время одиночного цикла 875 нс, в блочном режиме время цикла 750 нс. Работает почти со всеми ISA-адаптерами с большей скоростью.
- Type B — сокращенный на 50% цикл (750/500 нс на цикл), работает с большинством EISA-адаптеров и некоторыми ISA. Возможен только с памятью, непосредственно доступной контроллеру шины EISA (то есть с памятью на адаптерах EISA и системной, если EISA является основной шиной). Если декодированный адрес памяти относится к 8/16-битной памяти ISA, то контроллер DMA EISA автоматически переводится в режим Compatible.
- Type C (Burst Timing) — сокращенный на 87,5% цикл, ориентированный на пакетный режим передачи. Работает с быстрыми EISA-адаптерами. При обмене 32-битных устройств с 32-битной памятью позволяет развить скорость обмена до 33 Мбайт/с.
- В PCI-системах для обмена с устройствами системной платы (Fast ATA-2 или E-IDE-порты) возможно использование DMA Type F, при котором между соседними циклами интервал не должен превышать 3 тактов шины (360 нс). Для разгрузки системной шины используется 4-байтный буфер. Режим F может работать только в режиме одиночной передачи или по запросу и исключительно с инкрементом адреса. На самой шине PCI адаптеры могут использовать режим прямого управления шиной, для чего имеется специальный протокол арбитража.

[illegible]

Номер канала DMA#		0	1	2	3	4	5	6	7
Стандартное назначение	XT	MRFR*	–	FDD	HDD	Отсутствуют			
	AT	–	SDLC*	FDD	HDD*	Каскад	–	–	–
Разрядность, байт		1				2 с четного адреса			
Максимальный размер блока		64 Кбайта				128 Кбайт, четный			
Граница блока		Кратна 1000h				Кратна 2000h			
Регистр страниц		4 бит A16–A19				7 бит A17–A23			
Адреса регистров:									
страниц		087	083	081	082	08F	08B	089	087
начального адреса (W) текущего адреса (R)		000	002	004	006	0C0	0C4	0C8	0CC
начального счетчика (W) текущего счетчика (R)		001	003	005	007	0C2	0C6	0CA	0CE

* SDLC-адаптер устанавливается редко.

HDD-контроллер в AT DMA обычно не использует.

Канал 0 в XT используется для регенерации памяти (MRFR).

Канал 4 доступен только в PS/2 MCA.

А.5. Обмен данными

Открытая архитектура PC предрасполагает использование различных адаптеров расширения. Чаще всего ПО общается с адаптерами через порты ввода/вывода, используя прерывания, прямой доступ к памяти и непосредственное управление шиной. Связь через область памяти, к которой приписан адаптер, для «нештатных» адаптеров применяется редко. Выбор способа связи определяется пропускной способностью, временем отклика на события, происходящие в адаптере, и допустимой загрузкой процессора.

Программно-управляемый обмен подразумевает следующие действия:

- операцию чтения регистра состояния устройства для анализа его готовности;
- ожидание готовности (зацикливание предыдущего шага);
- обмен байтом или словом данных.

Такой обмен сильно загружает процессор, особенно если программа формирует управляющие сигналы обмена. Так, например, работает драйвер параллельного порта в стандартном режиме, когда строб данных формируется двумя инструкциями *OUT*. В результате пропускная способность такого порта в зависимости от процессора может быть порядка 150 Кбайт/с. Если разгрузить процессор от анализа готовности и формирования строба, как, например, в EPP-режиме параллельного порта, то производительность порта можно повысить на порядок. Такой режим обмена программно-управляемым называть некорректно — это *режим программного ввода/вывода с аппаратным контролем потока*, где темп обмена определяет подключенное устройство. Высокоскоростные *режимы PIO* применяются в интерфейсе IDE. Они используют инструкции блочной пересылки *REP INS/OUTS*, при которых процессор пересылает данные между портом и областью памяти за минимальное количество тактов системной шины. Поскольку такую скорость ПУ обычно не воспринимают, контроллер интерфейса «притормаживает» обмен до разумных скоростей, определяемых режимом обмена. Параметры этих режимов, называемых *PIO Mode*, приведены в табл. 6.10.

Обмен по прямому доступу к памяти (как и *режим прямого управления шиной*) в наименьшей степени загружает процессор — он занимается лишь анализом состояния и инициализацией канала DMA, но не передачей данных. Время отклика на запрос, когда контроллер «заряжен» на обмен, не превышает сотен наносекунд. Однако скорость стандартного канала DMA ограничена значением 2/4 Мбайт/с в зависимости от разрядности канала. Производительность в режимах прямого управления шиной обычно выше, чем у стандартных каналов DMA. Прогрессивные режимы DMA обеспечивают более высокие скорости обмена: Ultra DMA — до 33 Мбайт/с.

Обсудим теперь *инициализацию и синхронизацию*. Инициатором обмена выступает или ПО, или ПУ. Программа ожидает какого-либо события в ПУ, периодически читая его регистр состояния. Такой способ называется *обменом по опросу готовности*. Время реакции может составлять доли микросекунды, когда программа опрашивает устройство монопольно. Однако при этом процессор загружен бесполезной работой. Другой подход — использование аппаратных прерываний, вырабатываемых устройством по событиям, требующим внимания программы. Программные обработчики аппаратных прерываний инициализируют блочный обмен или выполняют одиночную операцию пересылки. Время реакции зависит от множества факторов, включая режим работы процессора. В защищенном режиме прерывание приводит к автоматическому сохранению контекста задачи в стеке и переключению задач. Эти действия связаны с интенсивным обменом с памятью, так что отклик может достигать десятков микросекунд или гораздо больше, если задействована виртуальная память. В реальном режиме процессора ответ на прерывание приходит за единицы микросекунд.

Возможно комплексное решение — опрос готовности устройств по периодическим прерываниям, например, от системного таймера — *polling*. Готовое устройство обслуживается, неготовое — пропускается до следующего прерывания. Процессор не выполняет бесполезных циклов опроса, а занимается другими задачами. Правда, расходы на обслужи-

вание прерываний остаются, а максимальное время реакции на событие не может быть меньшим, чем период таймера. Так работает утилита фоновой печати PRINT.

Активное использование прерываний характерно для многозадачных ОС.

А.6. Распределение системных ресурсов

Спецификация ISA требует, чтобы всем картам назначались свои системные ресурсы — области адресов в пространствах памяти и ввода/вывода, линии запросов прерываний и каналы прямого доступа к памяти. Платы не должны конфликтовать по ресурсам. Задача конфигурирования осложняется из-за отсутствия общего механизма автоматической передачи установленных параметров прикладному и системному ПО. Конфигурирование адаптеров выполняется переключением джамперов, затем установленные параметры заносятся в конфигурационные файлы.

Применение энергонезависимой памяти (NVRAM или ее разновидности — EEPROM), хранящей настройки, облегчает конфигурирование. С соответствующими адаптерами поставляется утилита настройки. Отсюда их названия: *Software Configured* (программно конфигурируемые) или *Jumperless* (без джамперов). Утилита может проверить выбираемые установки на отсутствие конфликта, однако достоверность обнаружения конфликта относительна. Некоторые программно конфигурируемые адаптеры все же имеют джамперы, что бывает полезно для установки типовой настройки, устраняющей конфликты (например, наложение областей памяти сетевого адаптера и видеопамяти). Преимуществом NVRAM является также отсутствие необходимости в конфигурационных файлах.

Ключевым моментом в автоконфигурировании является изоляция карты от остальных. Тогда ПО конфигурирования сможет вести с картой диалог, на который не влияет присутствие других устройств. Изоляция карт при конфигурировании заложена в шины MCA, PCI и EISA. В EISA для каждого слота возможно программное селективное управле-

ние сигналом AEN, запрещающим дешифрацию адресов портов ввода/вывода. В системе с шиной EISA имеется энергонезависимая память конфигурирования слотов. Необходимо также обеспечить единый метод двухстороннего обмена конфигурационной информацией между картой и ПО.

В шину PCI автоматическое конфигурирование установленных адаптеров заложено изначально. Здесь конфигурированию также подлежат *мосты шины* (*PCI Bridge*) — аппаратные средства подключения PCI к другим шинам. *Host Bridge* — главный мост — используется для подключения к системной шине. *Peer-to-Peer Bridge* — одноранговый мост — используется для соединения двух шин PCI (дополнительные шины PCI позволяют увеличить количество подключаемых устройств). При конфигурировании мостов указывается распределение системных ресурсов по шинам, которые они связывают. Таким образом, задаются пути транслирования управляющих сигналов по шинам и управление буферами данных, обеспечивая для каждого адреса памяти или ввода/вывода единственную шину назначения, по крайней мере для операций чтения (операции записи могут быть широковещательными). Подобная «маршрутизация» необходима и для сигналов запросов прерываний (каналы DMA к шине PCI отношения не имеют).

Среди устройств PnP (для шин ISA и PCI) выделяется класс динамически конфигурируемых устройств *DCD* (*Dynamically Configurable Device*). Ресурсы, используемые ими, динамически переназначаются, не требуя перезагрузки ОС. Если устройство DCD находится в *заблокированном состоянии* (*Locked DCD*), его ресурсы не могут быть изменены.

Полная поддержка PnP предусматривает наличие PnP BIOS, плат расширения PnP и/или модулей на системной плате. Для *хранения информации о системных ресурсах* необходимо было стандартизовать способы представления конфигурации. В декабре 1994 года компании Compaq, Intel и Phoenix опубликовали версию 1.03 документа «Extended System Configuration Data Specification» («Спецификация расширенных данных о системной конфигурации»), определяющего методы взаимодействия и структуры данных памяти для

конфигурации устройств *ESCD*. Данные ESCD хранятся в энергонезависимом хранилище информации *NVS* (Non-Volatile Storage). Это может быть память NVRAM или файл данных для ISA-систем, не имеющих поддержки PnP. NVRAM может отображаться на область (верхней) памяти, как полностью, так и постранично. Способ доступа к ESCD определяется вызовом специальной функции PnP BIOS.

ESCD разрабатывалась на основе форматов описания устройств шины EISA, в которой имеется специальная энергонезависимая память и утилиты конфигурирования *ECU* (EISA Configuration Utility). Все устройства описываются через *слоты* — специальные структуры данных. В стандарте EISA слот 0 используется для описаний устройств ISA, размещенных на системной плате. Слоты 1–15 соответствуют физическим слотам расширения шины EISA, слоты 16–64 определены для виртуальных (не связанных с конкретным физическим слотом шины) устройств. В ESCD сохранено примерно то же назначение слотов. Так устройства шины PCI (включая мосты) описываются слотами 16–64.

С данными ESCD взаимодействует PnP BIOS, а также ОС, поддерживающая PnP. *ACFG BIOS* (Auto-Configuration BIOS) имеет поддержку PnP, в частности автоконфигурирование. Конфигурированием устройств DCD, которые не были сконфигурированы ACFG BIOS во время POST, занимается менеджер конфигурирования, являющийся частью системного ПО. Каждая PnP-плата сообщает менеджеру конфигурирования о потребностях и возможных диапазонах настройки ресурсов. Для обычных (Legacy) плат ISA информация в ESCD заносится с помощью диалоговой утилиты конфигурирования *ICU* (ISA Configuration Utility). Таким образом, конфигурирование плат PnP выполняется автоматически, а обычных карт — джамперами или утилитами.

A.7. Спецификация PnP для шины ISA

Для изоляции карт ISA, программного распределения системных ресурсов, конфигурирования и передачи параметров ОС и прикладному ПО компаниями Intel и Microsoft была выработана спецификация «Plug and Play ISA Specification»;

версия 1.0а была опубликована в мае 1994 года. Рассмотрим реализацию PnP с точки зрения аппаратных средств.

Конфигурирование в системе PnP состоит из следующих шагов:

- Производится изоляция одной карты от всех остальных.
- Карте назначается *CSN* (Card Select Number).
- С карты считываются данные о сконфигурированных и поддерживаемых ресурсах.

Эти шаги повторяются для всех карт, после чего:

- производится распределение системных ресурсов, выделяемых каждой карте;
- каждая карта конфигурируется согласно выбранному распределению ресурсов и переводится в рабочий режим.

Все шаги конфигурирования выполняет POST, если BIOS имеет поддержку PnP, или ОС при загрузке. PnP BIOS может ограничиться конфигурированием и активацией устройств, участвующих в загрузке. BIOS без поддержки PnP может использовать необходимые для загрузки устройства, сконфигурированные с параметрами по умолчанию, а всем остальным занимается ОС. Конфигурирование выполняется в специальном состоянии плат, в которое их можно программно перевести специальным ключом инициализации, защищающим конфигурационную информацию от случайного разрушения.

Для конфигурирования карт PnP необходимы три 8-битных *системных порта* (табл. А.4), с которыми процессор общается по инструкциям ввода/вывода с однобайтной передачей данных. Карты PnP должны использовать 12-битное декодирование адреса ввода/вывода, а не 10-битное, как в традиционных картах ISA.

Порт *ADDRESS* используется для адресации регистров PnP, запись в него производится перед обращением к портам *WRITE_DATA* и *READ_DATA*. Он же применяется для записи последовательности кодов ключа инициализации. Выбор адреса для него обусловлен тем, что «разумные» карты расширения не будут использовать для записи адрес регистра состояния стандартного LPT-порта.

Порты *WRITE_DATA* и *READ_DATA* используются для обмена данными с регистрами PnP. Адрес порта *WRITE_DATA* традиционными картами с 10-битным декодированием будет восприниматься как адрес предыдущего порта, так что конфликт исключен. Перемещаемому адресу порта *READ_DATA* ПО PnP во время протокола изоляции может легко найти бесконфликтное положение. Адрес этого порта сообщается всем картам записью в их управляющий регистр PnP.

Таблица А.4. Системные порты ISA PnP

Имя порта	Адрес	Тип доступа
ADDRESS	0279h (*Printer status port*)	Только запись
WRITE_DATA	0A79h (*Printer status port* + 0800h)	Только запись
READ_DATA	Перемещаемый в диапазоне 0203h–03FFh	Только чтение

В конфигурационный режим логика PnP переводится *ключом инициализации* (*Initiation key*). Ключ представляет собой предопределенную последовательность записей в порт *ADDRESS*. Аппаратная логика карты, проверяющая ключ, основана на сдвиговом регистре с обратными связями *LFSR* (*Linear Feedback Shift Register*), схема которого приведена на рис. А.3. Во время проверки ключа на вход C1 подается уровень логического 0, а на вход C2 — стробы записи в порт *ADDRESS*. Логика, не показанная на рис. А.3, сравнивает код в сдвиговом регистре с текущей записью и при несовпадении сбрасывает *LFSR* в исходное состояние (код 6Ah). В это же состояние регистр переводится двумя последовательными записями нулей в порт *ADDRESS*. Сдвиг в регистре происходит при каждой записи в порт *ADDRESS*. Если ключ (последовательность из 32 записей требуемых байт) будет приложен верно, после последней записи логика карты перейдет в режим конфигурирования и подготовится к отработке протокола изоляции. Точная последовательность байт ключа в hex-формате выглядит следующим образом:

```
6A, B5, DA, ED, F6, FB, 7D, BE, DF, 6F, 37, 1B, 0D, B6, C3, 61,
B0, 58, 2C, 16, BB, 45, A2, D1, E8, 74, 3A, 9D, CE, E7, 73, 39.
```

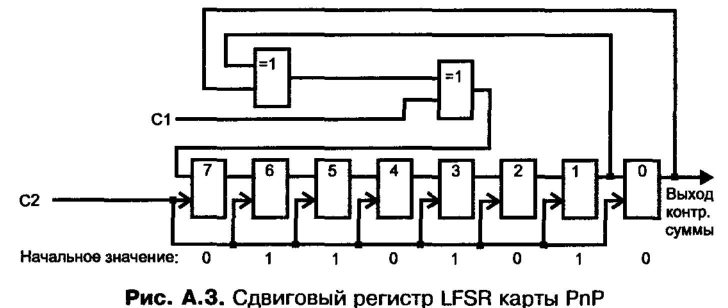


Рис. А.3. Сдвиговый регистр LFSR карты PnP

Протокол изоляции основан на идентификаторе *Serial Identifier*, хранящемся в памяти каждой карты PnP. Этот идентификатор представляет собой ненулевое 72-битное число, состоящее из двух 32-битных полей и 8-битного контрольного кода, вычисляемого с помощью регистра *LFSR*. Первое 32-битное поле представляет собой идентификатор производителя. Второе поле назначается производителем каждому экземпляру. Здесь может присутствовать серийный номер; для адаптера Ethernet это может быть частью MAC-адреса. Принцип построения идентификатора гарантирует, что в одной системе не могут встретиться две карты с совпадающими идентификаторами. Доступ к идентификатору осуществляется последовательно, начиная с бита 0 нулевого байта идентификатора производителя и заканчивая битом 7 контрольной суммы. Во время передачи идентификатора на вход C1 схемы *LFSR* поступают текущие биты идентификатора, а на вход C2 — стробы чтения регистра *Serial_Isolation*. В тактах передачи контрольной суммы ее биты берутся с выхода сдвигового регистра.

Протокол изоляции программно инициируется в любой момент времени посылкой ключа инициализации, переводящего все карты в конфигурационный режим. В этом режиме каждая карта ожидает 72 пары операций чтения порта *READ_DATA*. Ответ каждой карты на эти операции определяется значением очередного бита ее идентификатора.

Если текущий бит идентификатора карты имеет единичное значение, ее буфер шины данных в первом чтении пары выводит на шину значение 55h. Если текущий бит нулевой, то

буфер работает на *чтение* шины и логика карты анализирует ответ других карт — проверяет наличие комбинации 01 в битах D[1:0] (младшие биты 55h). В следующем цикле чтения пары карта с единичным битом выводит число AAh, а карта с нулевым текущим битом проверяет наличие комбинации 10 в битах D[1:0] (младшие биты AAh). Если карта, просматривающая вывод данных другими картами, обнаружила корректные коды в обоих циклах чтения пары, она в данной итерации изоляции исключается. Если карта в текущей паре управляла шиной или читала шину, но не обнаружила корректных активных ответов других карт, она сдвигает идентификатор на один бит и готовится к приему следующей пары циклов чтения. Эта последовательность выполняется для всех 72 бит идентификатора. В конце процесса останется лишь одна карта. Записью в управляющий регистр ей назначается номер *CSN*, по которому она будет использоваться в дальнейших операциях. Карта с назначенным *CSN* в следующих итерациях протокола изоляции не участвует (на пары чтений не отвечает).

Во время протокола изоляции карты не имеют права удлинять шинные циклы с помощью сигнала IOCHRDY, поскольку это привело бы к неопределенности результатов наблюдения за «соседями».

Программа конфигурирования проверяет данные, возвращаемые во время пар циклов чтения, и побитно собирает прочитанный идентификатор. Если в паре приняты байты 55h и AAh, соответствующий бит считается единичным, в других случаях — нулевым. При приеме идентификатора программа подсчитывает контрольную сумму и сравнивает ее с принятой. Несовпадение или отсутствие среди принятых байт 55h и AAh указывает на то, что выбранный адрес порта *READ_DATA* конфликтует с каким-либо устройством. Тогда программа производит итерацию, переместив адрес порта *READ_DATA* в допустимом диапазоне адресов. Если при переборе адресов не удается считать корректного идентификатора, принимается решение об отсутствии карт PnP в системе (вообще или с неназначенными *CSN*).

Программа должна обеспечить задержку 1 мс после подачи ключа перед первой парой чтений и 250 мкс между парами

чтений. Это дает карте время для доступа к информации, которая может храниться в медленных устройствах энергонезависимой памяти.

По завершении протокола изоляции ПО имеет список идентификаторов обнаруженных карт и присвоенных им номеров, сообщенных картам. Далее общение ПО с картой идет по ее номеру *CSN*, фигурирующему в командах PnP. Нулевой *CSN* присваивается картам по программному или аппаратному сбросу и используется как широковещательный адрес.

Обращения к регистрам PnP представляют собой операции записи/чтения портов ввода/вывода по адресам *WRITE_DATA/READ_DATA* соответственно. При этом для указания конкретного регистра PnP используется *индекс* — номер регистра, предварительно записанный в регистр *ADDRESS*.

Каждая карта имеет стандартный набор регистров PnP; часть из них относится к карте в целом, а часть — к логическим устройствам (ЛУ), входящим в карту. В любой момент времени в индексном пространстве регистров PnP отображаются общие регистры карты и регистры только одного ЛУ (рис. А.4). Выбор ЛУ, с которым производится общение, осуществляется записью в регистр *Logical Device Number*.

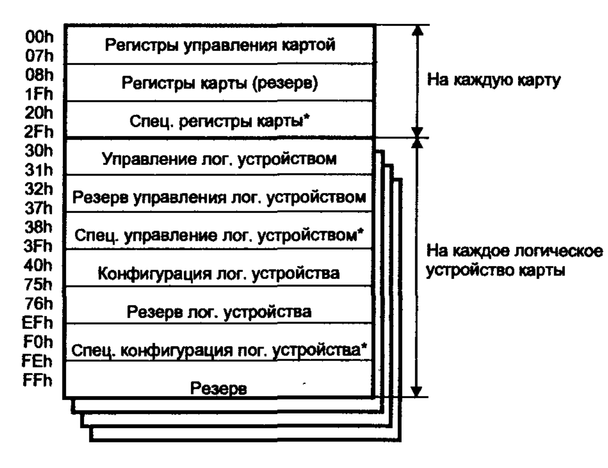


Рис. А.4. Конфигурационные регистры PnP (* — определяется разработчиком)

С точки зрения системы PnP карта пребывает в некотором состоянии. Состояние *Isolate* используется в протоколе изоляции, описанном выше.

Программирование карты начинается с команды ***WAKE[CSN]*** — записи байта ***CSN*** в регистр с индексом 3. Эта операция переводит карту с указанным ***CSN*** в состояние *Config* (конфигурирование), а остальные карты переходят в состояние *Sleep*. Для карты выполняются операции чтения ее конфигурационной информации (как карты в целом, так и ЛУ) и программирования используемых ресурсов. Программирование каждого ЛУ завершается установкой его бита активации, после чего оно активизируется на шине ISA. Программирование всей карты завершается переводом ее в состояние *Wait for key* (ожидание ключа). По окончании конфигурирования все карты PnP должны быть переведены в это состояние — тогда их случайное реконфигурирование будет блокировано 32-байтным ключом.

Доступ к регистрам PnP через ключ возможен в любое время. Возможно переназначение ***CSN*** «на ходу» — это требуется в устройствах, допускающих динамическое включение/выключение, док-станциях (*Docking Stations*) и при управлении энергопотреблением.

Все ЛУ карт PnP должны обеспечивать следующую минимальную функциональность:

- Регистры ресурсов при чтении должны отражать фактические текущие настройки.
- Бит активации при чтении должен отражать правдивое состояние активности устройства на шине ISA.
- Если программа пытается «навязать» карте неподдерживаемую конфигурацию, устройство не должно активироваться; соответственно, при чтении его флаг активации должен быть сброшен.

Стандартные регистры управления картой (табл. А.5) используются для адресации карты и ее ЛУ, а также для чтения конфигурационной информации (дескрипторов). Дескрипторы ресурсов могут быть считаны побайтно из регистра ***Resource Data***. При этом могут использоваться данные в короткой или

длинной форме. *Короткая форма* (*Small Resource Data Type*) допускает дескрипторы до 7 байт, *длинная форма* (*Large Resource Data Type*) — до 64 Кбайт. Форма дескриптора и имя описываемого ресурса определяются первым байтом дескриптора. Его последующие байты описывают требуемые ресурсы и возможные варианты (диапазоны) конфигурирования.

Таблица А.5. Стандартные регистры управления картой PnP

Регистр	Индекс и тип	Назначение
Set RD_DATA Port	00h, WO	Установка адреса порта для чтения. Биты [7:0] задают значение бит [9:2] адреса порта *READ_DATA*. Только для записи
Serial Isolation	01h, RO	Чтение этого регистра в состоянии *Isolation* приводит к побитному анализу идентификатора
Config Control	02h, WO	Бит [2] — сброс *CSN* в 0 Бит [1] — возврат в состояние *Wait for Key* Бит [0] — сброс всех ЛУ и перевод конфигурационных регистров в состояние включения питания, но *CSN* сохраняется. Эти биты не запоминаются, так что необходимости в их программном сбросе нет
Wake[CSN]	03h, WO	Запись в этот регистр приведет карту с записываемым байтом *CSN* к переходу из состояния *Sleep* в состояние *Config* (если данные не нулевые). Запись нулевого байта переводит все карты в состояние *Isolation*. Указатель последовательно считываемых байт сбрасывается
Resource Data	04h, RO	Чтение этого регистра возвращает очередной байт информации о ресурсах. Перед чтением должен опрашиваться регистр *Status*
Status	05h, RO	Регистр состояния. Единичное значение бита [0] указывает на возможность чтения очередного байта ресурсов

Таблица А.5 (продолжение)

Регистр	Индекс и тип	Назначение
Card Select Number	06h, RW	Регистр хранения селективного адреса *CSN*, обеспечивающего выбор конкретной карты командой *Wake[CSN]*
Logical Device Number	07h, RW	Выбор текущего ЛУ, к которому относятся все операции обмена конфигурационной информацией, проверки диапазона адресов ввода/вывода и активации. Если карта имеет одно устройство, регистр допускает только чтение и всегда имеет нулевое значение
Card Level Reserved	08h–1Fh	Зарезервированы
Card Level, Vendor Defined	20h–2Fh	Используются по усмотрению производителя

Таблица А.6. Стандартные регистры управления ЛУ PnP

Регистр	Индекс и тип	Назначение
Activate	30h, RW	*Регистр активации*: Бит [0] единичным значением разрешает активность ЛУ на шине ISA. Биты [7:1] зарезервированы, при чтении должны возвращать нули. Перед активацией проверка диапазона адресов ввода/вывода должна быть запрещена
I/O Range Check	31h, RW	*Регистр проверки диапазона адресов ввода/вывода*: Биты [7:2] зарезервированы, при чтении должны возвращать нули Бит [1] — разрешение проверки Бит [0] — управление диагностическим ответом: 0 — ответ AAh, 1 — ответ 55h
Logical Device Control Reserved	32h–37h	Зарезервированы
Logical Device Control Vendor Defined	38h–3Fh	Используются по усмотрению производителя

Стандартные регистры управления ЛУ (табл. А.6) используются для активации карт и проверки отсутствия конфликтов на шине ISA в выбранном диапазоне адресов ввода/вывода. Когда включен режим проверки конфликтов, на чтение по любому адресу установленного диапазона портов ввода/вывода ЛУ отвечает байтом 55h или AAh в зависимости от состояния бита 0 регистра проверки диапазона адресов. В рабочем режиме этот «автоответчик» отключен.

Оперативные данные конфигурирования доступны через регистры ЛУ. Каждое логическое устройство имеет собственные дескрипторы используемых системных ресурсов:

- Обычные 24-битные (4) и 32-битные (4) дескрипторы памяти. Поля базового адреса и длины неиспользуемого дескриптора памяти должны быть нулевыми. Одна карта не может одновременно использовать 24-битные и 32-битные дескрипторы памяти.
- Дескрипторы областей портов ввода/вывода (8). Поле базового адреса неиспользуемого дескриптора портов ввода/вывода должно быть нулевым. Размер области адресов определяется в блоке данных описателя.
- Дескрипторы запросов прерываний (2). Неиспользуемый селектор запроса прерывания должен быть нулевым (нулевой номер запроса занят системным таймером). Для линии IRQ2/9 шины ISA применяют номер 2.
- Дескрипторы каналов прямого доступа к памяти. Неиспользуемый дескриптор канала должен иметь значение 4 (этот канал недоступен — используется для каскадирования контроллеров).

Назначение регистров дескрипторов и их положение в индексном пространстве PnP описано в табл. А.7.

Таблица А.7. Дескрипторы системных ресурсов ЛУ

Индекс и тип	Назначение
40h–5Fh — *Обычные дескрипторы памяти*	
40h, RW	*Дескриптор памяти 0*: базовый адрес памяти, биты [23:16]

Индекс и тип	Назначение
41h, RW	*Дескриптор памяти 0*: базовый адрес памяти, биты [15:8]
42h, RW	*Дескриптор памяти 0*: управление. Бит [1] — режим обращения: 0 — 8 бит, 1 — 16 бит. Бит [0] (RO) — способ задания диапазона: 0 — следующее поле воспринимается как размер области, 1 — как старший адрес
43h, RW	*Дескриптор памяти 0*: размер или старший адрес области, биты [23:16]
44h, RW	*Дескриптор памяти 0*: размер или старший адрес области, биты [15:8]
45h–47h	*Заполнитель* (зарезервировано)
48h–4Ch	*Дескриптор памяти 1* (аналогично предыдущему)
4Dh–4Fh	*Заполнитель* (зарезервировано)
50h–54h	*Дескриптор памяти 2* (аналогично предыдущему)
55h–57h	*Заполнитель* (зарезервировано)
58h–5Ch	*Дескриптор памяти 3* (аналогично предыдущему)
5Dh–5Fh	*Заполнитель* (зарезервировано)
60h–6Fh — *Дескрипторы пространства ввода/вывода*	
60h, RW	*Дескриптор портов 0*: базовый адрес, биты [15:8]. Если ЛУ использует только 10-битное декодирование адреса, биты [15:10] могут игнорироваться
61h, RW	*Дескриптор портов 0*: базовый адрес, биты [7:0]
62h–63h, RW	*Дескриптор портов 1* — аналогично
64h–65h, RW	*Дескриптор портов 2* — аналогично
66h–67h, RW	*Дескриптор портов 3* — аналогично
68h–69h, RW	*Дескриптор портов 4* — аналогично
6Ah–6Bh, RW	*Дескриптор портов 5* — аналогично
6Ch–6Dh, RW	*Дескриптор портов 6* — аналогично
6Eh–6Fh, RW	*Дескриптор портов 7* — аналогично

Индекс и тип	Назначение
70h–73h — *Дескрипторы запросов прерываний*	
70h, RW	*Селектор запроса прерывания 0*: Биты [3:0] задают номер линии IRQ
71h, RW	*Тип сигнала запроса прерывания 0*: Бит[1] — активный уровень: 1 — высокий, 0 — низкий Бит[0] — тип: 1 — уровень, 0 — перепад Если карта поддерживает только один тип сигнала, регистр может быть типа RO
72h, RW	*Селектор запроса прерывания 1*
73h, RW	*Тип сигнала запроса прерывания 1*
74h–75h — *Дескрипторы каналов прямого доступа*	
74h, RW	*Селектор 0 канала DMA*. Биты [2:0] задают номер используемого канала (001 — DMA0, ... 111 — DMA7)
75h, RW	*Селектор 1 канала DMA*
76h–A8h — *32-битные дескрипторы памяти*	
76h, RW	*32-битный дескриптор памяти 0*: базовый адрес памяти, биты [31:24]
77h, RW	*32-битный дескриптор памяти 0*: базовый адрес памяти, биты [23:16]
78h, RW	*32-битный дескриптор памяти 0*: базовый адрес памяти, биты [15:8]
79h, RW	*32-битный дескриптор памяти 0*: базовый адрес памяти, биты [7:0]
7Ah, RW	*32-битный дескриптор памяти 0*: управление Биты [7:3] — зарезервированы, при чтении должны возвращать нули Биты [2:1] — управление доступом: 00 — 8-битная память, 01 — 16-битная память, 10 — зарезервировано, 11 — 32-битная память Бит [0] (RO) — способ задания диапазона: 0 — следующее поле воспринимается как размер области, 1 — следующее поле воспринимается как старший адрес

Индекс и тип	Назначение
7Bh, RW	*32-битный дескриптор памяти 0*: размер или старший адрес области, биты [31:24]
7Ch, RW	*32-битный дескриптор памяти 0*: размер или старший адрес области, биты [23:16]
7Dh, RW	*32-битный дескриптор памяти 0*: размер или старший адрес области, биты [15:8]
7Eh, RW	*32-битный дескриптор памяти 0*: размер или старший адрес области, биты [7:0]
7Fh	*Заполнитель* (зарезервировано)
80h–88h	*32-битный дескриптор памяти 1*
89h–8Fh	*Заполнитель* (зарезервировано)
90h–98h	*32-битный дескриптор памяти 2*
99h–9Fh	*Заполнитель* (зарезервировано)
A0h–A8h	*32-битный дескриптор памяти 3*

Приложение Б. Элементы цифровой схемотехники

В полупроводниковых цифровых микросхемах широко используются логические вентили на ТТЛ (TTL) и КМОП (CMOS) структурах. Внутри сложных микросхем применяются и другие типы ячеек, но они обычно обрамляются внешними схемами с параметрами ТТЛ- или КМОП-вентилей. Приведем некоторые свойства этих вентилей, знание которых полезно для работы с интерфейсами.

Логические микросхемы, применяемые в компьютерах, питаются от постоянного напряжения +5 В, приложенного относительно общего провода — шины GND. В современных компонентах (процессорах, микросхемах памяти) стремятся снизить напряжение питания до 3,3 В и ниже.

Существует несколько разновидностей микросхем ТТЛ. Стандартные микросхемы серий 74xxx имеют среднее потребление и быстродействие 10 нс, их отечественные аналоги — серии К155 и К133. Микросхемы с пониженным потреблением серии 74Lxxx и их аналоги К134 имеют пониженное быстродействие (33 нс). Серии 74Hxxx (К131), напротив, имеют повышенную выходную и потребляемую мощность. Микросхемы с диодами Шотки (ТТЛШ) 74Sxxx (К531) при более высоком, чем у стандартных, энергопотреблении имеют быстродействие в три раза выше (3 нс). Серия маломощных микросхем ТТЛШ 74LSxxx (К555) при том же быстродействии, что и у стандартной, потребляет мощность в несколько раз меньше. Наиболее перспективными являются серии 74Fxxx (КР1531) с быстродействием 3 нс и 74ALSxxx (КР1533) с быстродействием 4 нс. При этом потребление у серии ALS (Advanced Low-Power Schottky) в два раза ниже, чем у серии F (Fast). Серия ALS хорошо стыкуется с микросхемами КМОП.

В ТТЛ-логике различают входы, выходы (обычные, тристабильные и с открытым коллектором) и двунаправленные выводы.

Вход ТТЛ воспринимает только логический уровень сигнала. Порог переключения – обычно 1,3–1,4 В. Напряжение ниже порога воспринимается как низкий уровень, выше – как высокий. Состояние свободного (ни к чему не подключенного) входа ТТЛ-микросхемой воспринимается как высокоуровневое, и на нем высокоомным вольтметром или осциллографом можно наблюдать потенциал 1,3–1,4 В. В таком состоянии вход является чувствительным к помехам, поэтому свободные входы рекомендуют соединять с источником высокого или низкого логического уровня (в зависимости от логики работы). Если несколько свободных входов разных вентилей соединяются вместе, их состояние будет неопределенным: из-за разброса порогов часть из них может восприниматься как высокий уровень, а часть – как низкий. В качестве источника высокого уровня часто используют шину питания +5 В, но вход (или группу входов) подключают к ней через балластный резистор (1–10 кОм). В качестве низкого уровня используют общий провод (шину GND). Входной ток зависит от потенциала входа: при низком уровне ток имеет отрицательное значение (вытекающий ток) порядка 1,5 мА для стандартных микросхем ТТЛ, при высоком уровне – положительное (втекающий ток) на уровне десятков микроампер. У микросхем серий S, LS и ALS входные токи существенно меньше. Входное напряжение, превышающее значение питающего напряжения, для микросхем ТТЛ недопустимо – оно может пробить входной вентиль. Кроме вентилей с обычными входами существуют вентили с триггерами Шмитта. У них имеется гистерезис переключения около 0,8 В, симметричный относительно порога (1,3 В). Эти элементы используются как приемники сигналов с повышенным уровнем помех.

Обычный выход ТТЛ формирует выходные логические уровни: низкий (ниже 0,4–0,5 В) и высокий (выше 2,4 В). Выходные уровни при повышении нагрузки (выходного тока) ухудшаются – приближаются к порогу переключения. Выходной ток короткого замыкания (КЗ) на землю ограничен несколькими миллиамперами, поэтому КЗ на землю безопасно для выходов элементов ТТЛ. Выходной ток при КЗ на шину питания, когда вентиль пытается формировать низ-

кий уровень сигнала, достигает десятков миллиампер и опасен для микросхемы. Если два выхода соединить вместе и они будут пытаться формировать разные уровни, то в этом конфликте «победит» выход, формирующий низкий логический уровень. Этим свойством иногда пользуются при построении схем, но это не совсем «законно». Существуют буферные элементы с повышенной нагрузочной способностью. Они предназначены для подключения большого количества входов или цепей с большой емкостной нагрузкой. В PC таким местом является, например, мультиплексированная шина адреса динамической памяти.

Выход с открытым коллектором (*Open Drive Output*) работает в качестве ключа, способного коммутировать сигнал на шину GND. Этот тип выхода способен формировать только низкий логический уровень, а высокий уровень формируют с помощью внешнего резистора, «подтягивающего» сигнал к напряжению питания (*Pullup Resistor*). Выходы с открытым коллектором разрешается объединять, при этом реализуется функция «монтажное И». Существуют элементы с открытым коллектором, имеющие повышенную нагрузочную способность как по допустимому выходному току ключа, так и по допустимому напряжению на закрытом ключе. Они могут использоваться для управления исполнительными устройствами (например, реле), индикаторами и т. п.

Тристабильный выход (*Tristate Output*) кроме формирования низкого и высокого уровней может быть переведен в третье, высокоимпедансное (*High-Z State*) состояние, в котором выходной вентиль отключен от вывода. Этот тип выхода предназначен для объединения нескольких источников сигнала на одной шине. Как правило, не в третьем состоянии может находиться не более одного из объединяемых источников. В противном случае на шине будет конфликт, в котором побеждает низкий уровень. Вентили с тристабильным выходом имеют управляющий вход, который обычно обозначают OE (*Output Enable*).

Двунаправленный вывод элемента представляет собой комбинацию входа и тристабильного выхода (или выхода с открытым коллектором). В зависимости от управляющего сигнала этот вывод работает либо как вход, либо как выход.

Логические *элементы КМОП* отличаются от ТТЛ большим размахом сигнала (низкий уровень ближе к нулю, высокий — к напряжению питания), малыми входными токами (почти нулевыми в статике, в динамике — обусловленными паразитной емкостью) и малым потреблением, однако их быстродействие несколько ниже. В отличие от ТТЛ, микросхемы КМОП допускают более широкий диапазон питающих напряжений. Микросхемы ТТЛ и КМОП взаимно стыкуются, хотя вход КМОП требует более высокого уровня логической единицы, а выход КМОП из-за невысокого выходного тока можно нагружать лишь одним ТТЛ-входом. Современные микросхемы КМОП по параметрам приближаются к ТТЛ серии ALS и хорошо стыкуются с ними. Микросхемы КМОП имеют те же типы выводов, но вместо выхода с открытым коллектором у них присутствует выход с открытым стоком (что по логике работы одно и то же).

Длина интерфейсных кабелей ограничивается как уровнем помех на входе, так и создаваемой емкостной нагрузкой на выходные вентили, в качестве которых рекомендуется применять элементы с повышенной нагрузочной способностью. Длина кабелей *Centronics* ограничена несколькими метрами, в то время как для интерфейса *RS-232C* допустимы кабели длиной в десятки метров (сказывается большая разница уровней и зона нечувствительности).

Логические схемы могут быть чисто комбинационными вентилями (*Gate*), у которых состояние выходов определяется только текущим состоянием входов, или элементами с памятью. В схемах последнего типа состояние выхода определяется предысторией входных сигналов и внутренних состояний. К ним относятся разнообразные триггеры, регистры, счетчики и т. п. Элементы могут быть асинхронными и синхронными. У последних состояние входов стробируется потенциалом или перепадом на специальном входе синхронизации. Следует особо отметить два типа элементов, широко применяемых в микропроцессорной технике. *Регистром* (*Register*) называют совокупность нескольких запоминающих элементов, запись в которые производится по общему управляющему сигналу. Подразумевается, что в регистре информация воспринимается по перепаду сигнала синхрони-

зации (на рис. Б.1а запись происходит по положительному перепаду). *Защелкой* (*Latch*), или регистром-защелкой, называют схему, работающую иначе (рис. Б.1б). Здесь при одном состоянии управляющего входа (высоком) регистр «прозрачен» — на выходе отражаются изменения на входах, а при переходе этого сигнала в другое состояние на выходах фиксируется состояние, присутствующее к этому моменту. Защелки используются для фиксации адреса на шине микропроцессора, позволяя схемам дешифраторов адреса раньше начинать работу, тем самым сокращая затраты времени на дешифрацию адреса. До срабатывания на выходе защелки возможен «мусор» от переходных процессов на входе, чего не бывает в регистрах, синхронизируемых по перепаду.

Для того чтобы любая синхронизируемая схема зафиксировала желаемое состояние, сигналы на входах должны установиться до синхронизирующего перепада за некоторое время, называемое временем установки T_{SETUP}, и удерживаться после него в течение времени удержания T_{HOLD}. Значение этих параметров определяется типом и быстродействием синхронизируемой схемы, и в пределе один из них может быть нулевым.

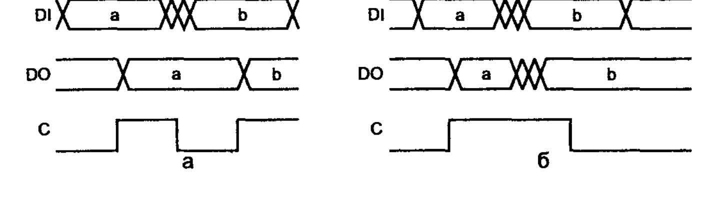


Рис. Б.1. Диаграмма работы регистров:
а — регистр, *б* — регистр-защелка

Наконец, рассмотрим типовое подключение некоторой функциональной микросхемы (например, i8255 — КР580ВВ55) к шине ISA, которое хорошо иллюстрирует принципы сопряжения устройств микропроцессорной техники (рис. Б.2). Для сопряжения с микропроцессором имеется *шина данных* (*Data Bus*), *шина адреса* (*Address Bus*) и *шина управления* (*Control Bus*). Первые две из них могут использовать одни и те же физические линии, такое решение называется *мультиплек-*

сированием шины адреса и данных. Тогда в шине управления будет присутствовать сигнал, определяющий назначение шины в данный момент времени. В нашем примере шины адреса и данных разделены.

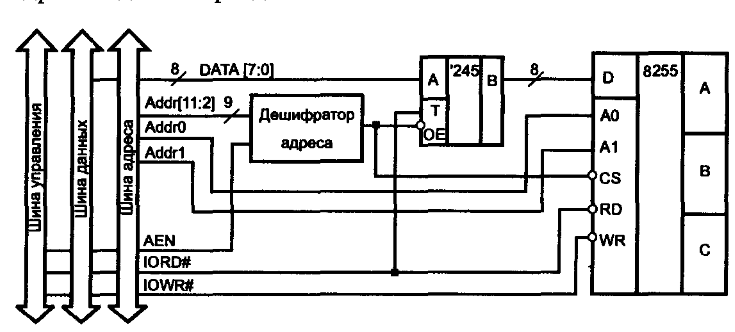


Рис. Б.2. Подключение устройства к шине ISA

Подключаемое устройство обычно имеет свой буфер данных — двунаправленный приемопередатчик, в качестве которого применяется микросхема 74ALS245 (1533АП6). Буфер должен открываться сигналом OE# (Output Enable — разрешение выхода), когда на шине адреса присутствует адрес, относящийся к диапазону подключаемого устройства. «Дежурным» направлением передачи является «от шины — к устройству»; переключение в обратную сторону производится по сигналу IORD# шины управления. Таким образом, буфер имеет право передавать данные на шину (управлять шиной данных) только во время действия сигнала чтения, относящегося к зоне адресов данного устройства. Если бы подключаемое устройство было приписано к пространству памяти, в логике управления направлением присутствовал бы сигнал MEMRD#.

Дешифратор адреса предназначен для выявления зоны адресов, относящейся к подключаемому устройству. Если устройству требуется более одного адреса, младшие линии адреса (в данном примере — A0 и A1) используются для декодирования адреса внутри устройства. Остальные линии поступают на вход комбинационной схемы (или программируемой логической матрицы), которая формирует сигнал

обращения к устройству, называемый CS# (*Chip Select*). На шинах ISA срабатывание дешифратора адреса должно блокироваться высоким уровнем сигнала AEN, сигнализирующим о недействительности адреса для порта ввода/вывода в цикле DMA.

Шина управления представлена сигналами IORD# (чтение порта), IOWR# (запись в порт) и AEN. Их состав может быть расширен сигналами обращения к памяти MEMRD# и MEMWR#, а также сигналами запросов прерываний, управления каналами прямого доступа и др. Приведенные четыре сигнала обращения к портам и памяти, используемые в шине ISA, характерны для микропроцессорных наборов и периферийных схем i8080. Есть другой набор сигналов — в стиле i8085: сигнал M/IO# определяет, к чему относится обращение — к памяти (M) или вводу/выводу (IO), сигнал W/R# определяет тип операции — запись (W) или чтение (R), а сама операция осуществляется по некоторому синхронизирующему сигналу. Такой способ применяется в шинах EISA и PCI.

В микропроцессорной технике применяются прямые и инверсные логические сигналы. В случае прямых сигналов логическому нулю соответствует низкий уровень сигнала, логической единице — высокий. В случае инверсных сигналов все наоборот. Инверсию сигналов обозначают разными способами: перед названием сигнала ставят знак «минус», над именем проводят черту, после имени ставят обратную косую черту или решетку. В данной книге используется последний способ. Управляющие сигналы обычно инверсные. Это так называемые *L*(*Low*)-активные сигналы, у которых активный уровень сигнала — низкий. Это нужно, чтобы:

- повысить помехозащищенность, которая у ТТЛ несимметричная. Входные токи стремятся подтянуть уровень к высокому, и в случае прямых *H*(*High*)-активных сигналов это действует согласно с помехой, чреватой ложными срабатываниями. При L-активных сигналах входной ток противодействует помехе. Особенно важно использовать L-активность для сигналов, передаваемых по кабелям.

- обеспечить возможность нескольким источникам управлять одной и той же линией. L-активная линия «подтягивается» к высокому уровню резистором, а активный сигнал может вводить любой подключенный к ней вентиль с открытым коллектором (можно с тристабильным выходом).

В IBM PC принцип L-активности управляющих сигналов интерфейса был нарушен дважды: H-активность имеют сигналы запросов аппаратных прерываний IRQx и каналов прямого доступа DRQx. Это привело к невозможности совместного использования линий прерываний и каналов DMA.

Обозначение и порядок бит и байт шин адреса/данных пришло от процессоров Intel 8086/88. В шине данных D0 обозначает самый младший бит *LSB* (*Least Significant Bit*), а D7 — старший бит байта — *MSB* (*Most Significant Bit*). Иногда в описании интерфейсов биты данных обозначаются как D1...D8, при этом младший бит — D1. На рисунках принято старший бит изображать слева, а младший — справа. Обозначение D[7:0] относится к группе сигналов D7, D6,..., D1, D0, а D[0:7] — к тем же сигналам, но в порядке естественной нумерации. В двухбайтном слове, размещаемом в памяти, принят *LH-порядок* следования: адрес слова указывает на младший байт *L* (*Low*), а старший байт *H* (*High*) размещается по адресу, на единицу большему. В двойном слове порядок будет аналогичным — адрес укажет на самый младший байт, после которого будут размещены следующие по старшинству. Этот порядок естествен для процессоров Intel.

В цифровой схемотехнике есть множество тем для обсуждения, остановимся на том, что уже изложено.

Приложение В. Конструктивные элементы интерфейсов

Определим некоторые термины, относящиеся к аппаратным средствам современных компьютеров.

Системной (System Board), или *материнской, платой* (Mother Board) называют основную печатную плату, на которую устанавливают процессор, оперативную память, ROM BIOS и другие системные компоненты.

Платой или *картой расширения* (Expansion Card) называют печатную плату с краевым разъемом, устанавливаемую в слот расширения. Карты расширения, обеспечивающие какой-либо дополнительный интерфейс, называют *интерфейсными картами* (Interface Card). Их также называют *адаптерами* (Adapter). К примеру, дисплейный адаптер (Display adapter) служит для подключения монитора.

Слот (Slot) представляет собой щелевой разъем, в который устанавливается какая-либо печатная плата. *Слот расширения* (Expansion Slot) в PC представляет собой разъем системной шины в совокупности с прорезью в задней стенке корпуса компьютера — то есть посадочное место для установки карты расширения. Слоты расширения имеют разъемы шин ISA/EISA, PCI, AGP, MCA, VLB или PC Card (PCMCIA). Внутренние слоты используются для установки модулей оперативной памяти (DIMM), кэш-памяти (COAST), процессоров Pentium II и т. д.

Сокет (Socket) — гнездо, в которое устанавливаются микросхемы. Его контакты рассчитаны на микросхемы со штырьковыми выводами в корпусах DIP и PGA во всех модификациях или же микросхемы в корпусах SOJ и PLCC с выводами в форме буквы «J». *ZIF-Socket* (Zero Insertion Force — с нулевым усилием вставки) предназначен для легкой установки при высокой надежности контактов. Эти гнезда имеют замок, открыв который можно установить или

изъять микросхему без приложения усилия к ее выводам. После установки замок закрывают, при этом контакты сокета плотно обхватывают выводы микросхемы.

Джампер (Jumper) — съемная перемычка, устанавливаемая на торчащие из печатной платы штырьковые контакты (рис. В.1а). Джамперы используются для конфигурирования различных компонентов, которые не требуют оперативного управления. Джамперы переставляют с помощью пинцета при выключенном питании.

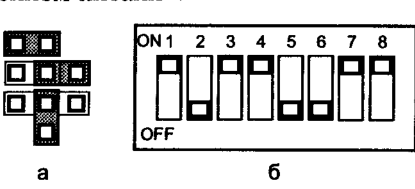


Рис. В.1. Аппаратные средства конфигурирования: а — джампер, *б* — DIP-переключатель

DIP-переключатели (DIP Switch) — малогабаритные выключатели в корпусе DIP (рис. В.1б), применяемые для тех же целей, что и джамперы. Более легки в переключении. Недостатками являются большее занимаемое на плате место и более высокая цена. Обычно являются только выключателями, что делает их применение менее гибким по сравнению с джамперами.

Платы (карты), в которых нет джамперов, называют *Jumperless Cards.* Компоненты, которые после установки конфигурируются автоматически, относят к классу *PnP* (Plug and Play — вставляй и играй).

Чип (Chip) — полупроводниковая микросхема. *Чипсет* (Chip Set) — набор специализированных интегральных схем, при подключении которых друг к другу формируется функциональный блок вычислительной системы. Чипсеты применяются в системных платах, графических контроллерах и других устройствах, функции которых нельзя реализовать в одной микросхеме.

Для соединения устройств и узлов PC применяют различные *разъемы,* они же *коннекторы* (Connector). Среди них чаще всего встречаются следующие:

- Разъемы *D-типа* (рис. В.2) используются для подключения внешних устройств — мониторов, принтеров, модемов и т. д. Розетки (Female, «мамы») обозначаются как *DB-xxS*, где xx — количество контактов. Вилки (Male, «папы») обозначаются как *DB-xxP*. Ключом является D-образный кожух. Назначение разъемов, выходящих на заднюю стенку PC, стандартизовано (табл. В.1).

DB-25S DB-25P DB-15S (Game) DB-9S DB-9P DB-15S (VGA)

Рис. В.2. Разъемы D-типа (вид с наружной стороны)

Таблица В.1. Назначение разъемов D-типа

Тип разъема	Назначение
DB-9P (вилка)	COM-порт
DB-9S (розетка)	Выход на монитор (Mono, CGA, EGA)
DB-15S (розетка) двухрядный	Game-порт, MIDI
DB-15S (розетка) трехрядный	Выход на монитор (VGA/SVGA)
DB-25P (вилка)	COM-порт
DB-25S (розетка)	LPT-порт

- Разъемы *IDC* (Insulation-Displacement Connector — разъем, смещающий изоляцию) получили свое название из-за способа присоединения кабеля. Контакты разъема со стороны, обращенной к кабелю, имеют ножи, подрезающие и смещающие изоляцию проводников кабеля. Разъемы предназначены для ленточных кабелей шлейфов, хотя возмож-

на заделка в них и одиночных проводников. Для заделки кабелей в эти разъемы существуют специальные инструменты – прессы. Разъемы IDC существуют для краевых печатных разъемов (рис. В.3а) и штырьковых контактов (рис. В.3б). Разъемы могут иметь ключи: для печатных разъемов это прорезь и соответствующая ей перемычка, расположенная ближе к первым контактам. Для штырьковых разъемов ключом является выступ на корпусе, но этот ключ сработает, только если ответная часть имеет пластмассовый бандаж с прорезью. Ключом может являться отсутствующий штырек — на разъеме для него не оставляют отверстия (такой ключ рекомендуется стандартом ATA). На ленточном кабеле крайний провод, соединяемый с контактом «1», маркируют цветной краской. На печатной плате штырек «1» обычно имеет отличающуюся от других (квадратную) форму контактной площадки. Разъемы IDC и ленточные кабели-шлейфы применяют для подключения внешних разъемов к системной плате и картам расширения и для подключения накопителей.

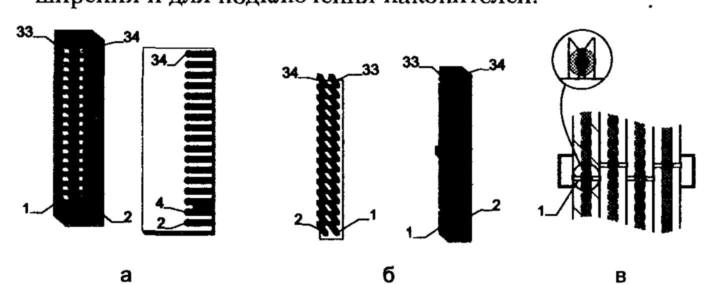


Рис. В.3. Разъемы IDC:
а — краевые, *б* — штырьковые, *в* — заделка проводов

- Разъемы типа *Centronics* (рис. В.4) применяют на принтерах и внешних устройствах SCSI.

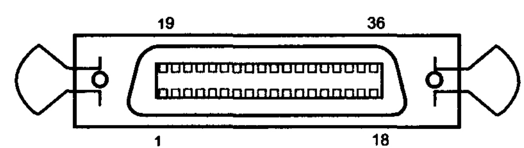


Рис. В.4. Разъемы типа Centronics

В интерфейсах применяют кабели различных типов — экранированные, неэкранированные, с витыми парами проводов, плоские кабели-шлейфы и т. п. С точки зрения частотных параметров и помехозащищенности лучшим способом является передача каждого сигнала в дифференциальном виде по отдельной витой паре проводов, но это дорого. Неплохой результат дает линейная (обычная) передача сигнала, но так, чтобы сигнальный провод был перевит с собственным обратным проводом, соединенным с шиной GND на обоих концах интерфейса. Чуть хуже, но дешевле использование плоского кабеля-шлейфа, в котором сигнальные проводники чередуются с «землей» или используются дифференциальные пары. Как правило, чем длиннее соединительный кабель, тем ниже его пропускная способность. Поэтому, если с длинным кабелем возникают проблемы, надо либо менять кабель на более качественный и/или короткий, либо снижать физическую скорость обмена. В маркировке проводов кабелей часто фигурирует обозначение вида 24 AWG. Оно определяет сечение проводника согласно стандарту AWG (American Wire Gauge), как показано в табл. В.2.

Таблица В.2. Классификация проводов по AWG

Номер по AWG	Диаметр, мм	Сечение, $мм^2$	Сопротивление 1 нм провода, Ом	Допустимый тон, А*
46	0,04	0,0013	13700	0,0038
44	0,05	0,0020	8750	0,006
42	0,06	0,0028	6070	0,009
41	0,07	0,0039	4460	0,012
40	0,08	0,0050	3420	0,015
39	0,09	0,0064	2700	0,019
38	0,10	0,0078	2190	0,024
37	0,11	0,0095	1810	0,028
—"—	0,12	0,011	1520	0,033
36	0,13	0,013	1300	0,040
35	0,14	0,015	1120	0,045
—"—	0,15	0,018	970	0,054

Таблица В.2 (*продолжение*)

Номер по AWG	Диаметр, мм	Сечение, $мм^2$	Сопротивление 1 км провода, Ом	Допустимый ток, А*
34	0,16	0,020	844	0,06
—"—	0,17	0,023	757	0,068
33	0,18	0,026	676	0,075
—"—	0,19	0,028	605	0,085
32	0,20	0,031	547	0,093
30	0,25	0,049	351	0,147
29	0,30	0,071	243	0,212
27	0,35	0,096	178	0,288
26	0,40	0,13	137	0,378
25	0,45	0,16	108	0,477
24	0,50	0,20	87,5	0,588
—"—	0,55	0,24	72,3	0,715
—"—	0,60	0,28	60,7	0,85
22	0,65	0,33	51,7	1,0
—"—	0,70	0,39	44,6	1,16
—"—	0,75	0,44	38,9	1,32
20	0,80	0,50	34,1	1,51
—"—	0,85	0,57	30,2	1,70
19	0,90	0,64	26,9	1,91
—"—	0,95	0,71	24,3	2,12
18	1,00	0,78	21,9	2,36
—"—	1,10	0,95	18,1	2,85
—"—	1,20	1,1	15,2	3,38
16	1,30	1,3	13,0	3,97
—"—	1,40	1,5	11,2	4,60
—"—	1,50	1,8	9,70	5,30
14	1,60	2,0	8,54	6,0
—"—	1,70	2,3	7,57	6,7
13	1,80	2,6	6,76	7,6
—"—	1,90	2,8	6,05	8,5
12	2,00	3,1	5,47	9,4

* При допустимой плотности тока 3 $А/мм^2$.

Приложение Г. Проблемы заземления

Вопросы электропитания играют важную роль в устойчивости работы компьютеров, их сетей и ПУ, соединяемых интерфейсами, а также в обеспечении их долголетия. Понимание некоторых вопросов электротехники позволит обойтись без «пиротехнических эффектов» при соединении устройств. Рассмотрим правила подключения к питающей сети с точки зрения безопасности как человека, так и компьютера.

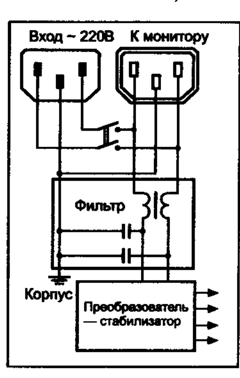


Рис. Г.1. Входные цепи блока питания

Практически каждый блок питания компьютера или ПУ имеет сетевой фильтр (рис. Г.1). Конденсаторы этого фильтра предназначены для шунтирования высокочастотных помех питающей сети на землю через провод защитного заземления и соответствующую трехполюсную вилку и розетку. «Земляной» провод соединяют с контуром заземления, но допустимо его соединять и с «нулем» силовой сети (разница ощущается только в особо тяжелых условиях эксплуатации). При занулении необходимо *быть уверенным в том, что «нуль» не станет фазой*, если кто-нибудь вдруг перевернет вилку питания. Если же «земляной» провод устройства никуда не подключать, на корпусе устройства появится *напря-*

жение порядка 110 В переменного тока (рис. Г.2): конденсаторы фильтра работают как емкостной делитель напряжения, и поскольку их емкость одинакова, 220 В делится пополам.

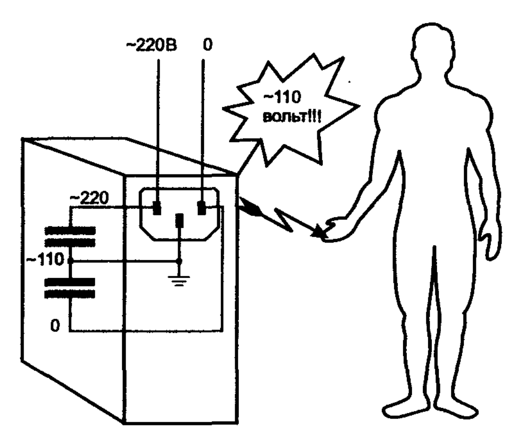


Рис. Г.2. Образоаание потенциала на корпусе компьютера

Конечно, мощность этого «источника» ограничена — ток короткого замыкания $I_{к.з}$ на землю составляет от единиц до десятков миллиампер, причем, чем мощнее блок питания, тем больше емкость конденсаторов фильтра и, следовательно, ток:

$I_{к.з} = U_{пит} \times 2\pi FC$,

где $U_{пит}$ = 220 В, F = 50 Гц — частота питающей сети, С — емкость конденсатора фильтра. При емкости конденсатора С = 0,01 мкФ этот ток будет около 0,7 мА.

Такие напряжение и ток *опасны для человека*. Попасть под напряжение можно, прикоснувшись одновременно к неокрашенным металлическим частям корпуса компьютера и, например, к батарее отопления. Это напряжение является одним из источников разности потенциалов между устройствами, от которой страдают интерфейсные схемы.

Посмотрим, что происходит при соединении двух устройств (компьютера и принтера) интерфейсным кабелем. Общий провод интерфейсов последовательных и параллельных портов связан со «схемной землей» и корпусом устройства. Если

соединяемые устройства надежно *заземлены* (занулены) *через отдельный* провод на общий контур (рис. Г.3), *проблемы разности потенциалов не возникает.*

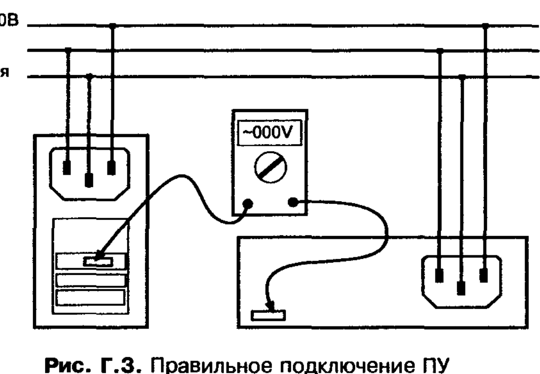


Рис. Г.3. Правильное подключение ПУ

Если же в качестве заземляющего провода использовать нулевой провод питания при разводке питающей сети с трехполюсными розетками *двухпроводным* кабелем, на нем будет набегать *разность потенциалов*, вызванная падением напряжения от протекающего силового тока I_{NUL} (рис. Г.4).

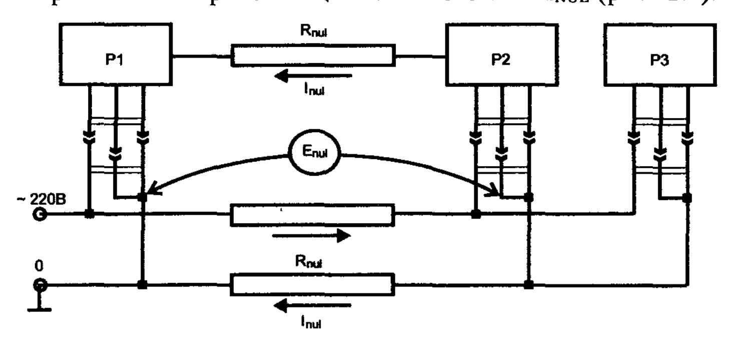


Рис. Г.4. Появление разности потенциалов при двухпроводном кабеле питания

Если в эти же розетки включать устройства с большим энергопотреблением, разность потенциалов (и импульсные помехи при включении-выключении) будет ощутимой. При этом эквивалентный источник напряжения при относительно невысокой ЭДС. E_{nul} (несколько вольт) будет иметь очень

низкое выходное сопротивление, равное сопротивлению участка нулевого провода (доли Ом). Уравнивающий ток через общий провод интерфейса I_{INT} можно оценить по формуле

$I_{int} = E_{nul}/(R_{nul} + R_{int})$,

где $E_{nul} = I_{nul} \times R_{nul}$, $I_{nul} = P/220$, R_{nul} — сопротивление нулевого провода и соединительных контактов розеток, R_{int} — сопротивление общего провода интерфейса, P — мощность, потребляемая устройствами, расположенными на рис. Г.4 справа (P = P2 + P3).

Поскольку обычно сопротивление интерфейсного кабеля больше питающего, через общий провод интерфейса потечет ток существенно меньший, чем силовой. Но при нарушении контакта в нулевом проводе питания через интерфейсный провод может протекать и весь ток, потребляемый устройством. Он может достигать нескольких ампер, что повлечет выход устройств из строя. Невыровненные потенциалы корпусов устройств также являются источником помех в интерфейсах.

Если оба соединяемых устройства не заземлены, в случае их питания от одной фазы сети разность потенциалов между ними будет небольшой (вызванной разбросом емкостей конденсаторов в разных фильтрах). Уравнивающий ток через общий провод интерфейса будет мал, и разность потенциалов между схемными землями устройств тоже будет мала. Но не следует забывать о безопасности человека. Если незаземленные устройства подключены к разным фазам, разность потенциалов между их несоединенными корпусами будет порядка 190 В, при этом уравнивающий ток через интерфейс может достигать десятка миллиампер. Когда все соединения/разъединения выполняются при отключенном питании, для интерфейсных схем такая ситуация почти безопасна. Но при коммутациях при включенном питании возможны неприятности: если контакты общего провода интерфейса соединяются позже (разъединяются раньше) сигнальных, разность потенциалов между схемными землями прикладывается к сигнальным цепям, и они выгорают. Самый тяжелый случай — соединение заземленного устройства с незаземленным (рис. Г.5), особенно когда у последнего мощный блок питания.

Для устройств, блоки питания которых имеют шнуры с двухполюсной вилкой, эти проблемы тоже актуальны. Такие блоки питания зачастую имеют сетевой фильтр, но с конденсаторами малой емкости (ток короткого замыкания достаточно мал).

Весьма *коварны* сетевые *шнуры компьютеров с двухполюсной вилкой*, которыми подключаются блоки питания с *трехполюсным разъемом*. Пользователи, подключающие свои компьютеры в бытовые розетки, могут столкнуться с проблемами из-за отсутствия заземления.

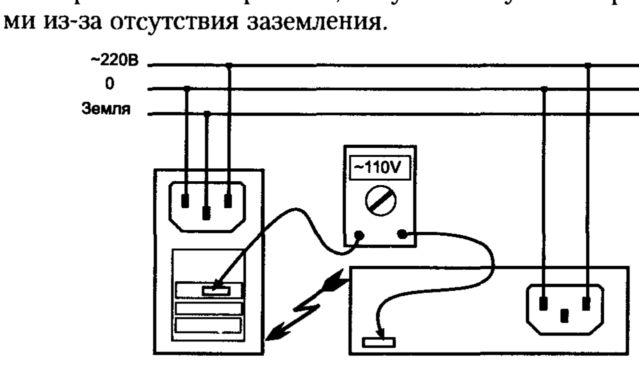


Рис. Г.5. Подключение незаземленного устройства

Локально проблемы заземления решает применение сетевых фильтров типа Pilot и им подобных. Питание от одного фильтра всех устройств, соединяемых интерфейсами, решает проблему разности потенциалов. Еще лучше, когда этот фильтр включен в трехполюсную розетку с заземлением (занулением). Однако заземляющие контакты (обжимающие «усики») многих розеток могут иметь плохой контакт вследствие своей слабой упругости или заусениц в пластмассовом кожухе. Кроме того, эти контакты не любят частого вынимания и вставки вилок, так что обесточивание оборудования по окончании работы лучше выполнять выключателем питания фильтра (предварительно выключив устройства).

Настоятельно рекомендуется *отключать питание при подключении и отключении интерфейсных кабелей*. Небольшая разность потенциалов, которая практически исчезнет при соединении устройств общими проводами интерфейсов, может пробить входные (и выходные) цепи сигнальных линий, если в момент присоединения разъема контакты общего провода соединятся позже сигнальных. От такой последовательности обычные разъемы не страхуют.

К помехам, вызванным разностью потенциалов схемных земель (корпусов) устройств, наиболее чувствительны параллельные порты. У последовательных портов зона нечувствительности шире (пороги ±3 В), еще меньшую чувствительность имеют интерфейсы локальных сетей, где обычно имеется гальваническая развязка сигнальных цепей от схемной земли с допустимым напряжением изоляции порядка 100 В.

Проблема заземления устройств, сильно *разнесенных территориально*, обостряется. Если разводка питания и заземления выполнена двухпроводным кабелем (см. рис. Г.4), разность потенциалов, обусловленная падением напряжения на заземляющих проводах, будет особенно ощутимой. В ряде случаев практикуется прокладка отдельного кабеля или шины для цепи заземления. Однако разводка заземления отдельным кабелем не всегда удобна и часто неэффективна с точки зрения защиты от помех, поскольку при этом могут образовываться замкнутые контуры с широким охватываемым пространством — своеобразные антенны. Так что разводку питания к устройствам целесообразно выполнять трехпроводным кабелем, один из проводов которого используется для защитного заземления. При этом древовидная схема заземления получается естественным образом (рис. Г.6), защитный провод в корневой части этого дерева заземляют или зануляют. Все устройства, электрически соединяемые между собой, *желательно питать от одной фазы сети*, хотя, с точки зрения энергетиков, это требование часто трудновыполнимо.

Дополнительные проблемы при разводке электропитания для компьютеров обусловлены ярко выраженной динамической нелинейностью входной цепи бестрансформаторных блоков питания. Традиционные электросети рассчитаны на более или менее линейную нагрузку, у которой в спектре тока основная мощность приходится на первую гармонику. В трехфазной сети с равномерно распределенной по фазам линейной нагрузкой в идеале через нейтральный провод ток практически не течет, поскольку токи от нагрузок всех трех фаз компенсируют друг друга. Учитывая это свойство, во многих четырехпроводных кабелях сечение проводника для нейтрали существенно меньше, чем сечение фазных проводников. При нелинейной симметричной нагрузке фаз при большом

уровне третьей гармоники тока (что характерно для бестрансформаторных блоков питания) взаимной компенсации токов не происходит, и действующее значение тока в нулевом проводе оказывается даже больше, чем в каждом из фазных.. Таким образом, при подключении большого числа компьютеров к традиционной 4-проводной трехфазной проводке происходит перегрузка нулевого провода. Эта перегрузка приводит к следствиям разной степени тяжести — от «набегания» помехи переменного тока на нулевом проводе до перегорания нулевого провода, который никогда не защищают от перегрузки — все автоматы защиты ставятся только в фазных проводах. Во избежание перегрузки нулевого провода и в случае питания от трехфазной сети силовую разводку к розеткам от распределительного щита следует опять-таки вести трехпроводным кабелем. Перегрузки нулевого провода подводящего силового кабеля можно избежать, установив в распределительном щите развязывающий трехфазный трансформатор 380/220 В. К этому трансформатору входное напряжение подводится по схеме «треугольник», а выходные обмотки соединяют по схеме «звезда».

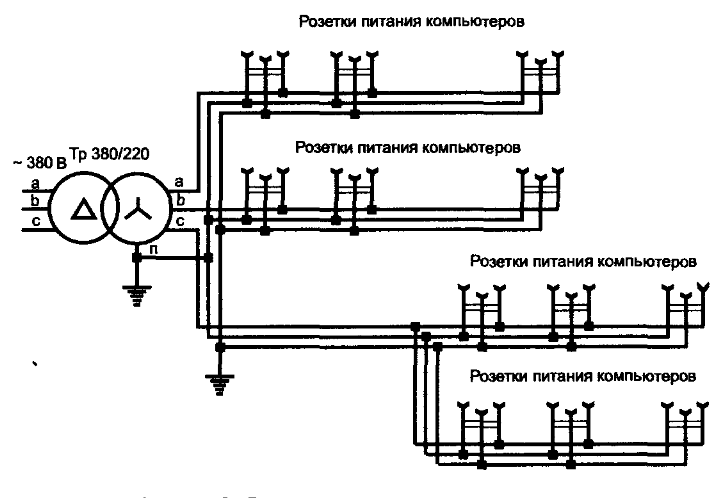


Рис. Г.6. Разводка питания и заземления

Список сокращений

8041/8042	ИС контроллера клавиатуры
8237	ИС контроллера DMA
8250, 16450, 16550	ИС последовательного интерфейса (UART)
8253/8254	ИС трехканального таймера PC
8255	ИС программируемого параллельного интерфейса (PIO) и системного порта PC XT
8259A	ИС контроллера прерываний (PIC)
ACCESS.Bus	Accessory Bus, шина подключения ПУ
ADC	Analog to Digital Converter, аналого-цифровой преобразователь
ADPCM	Adaptive Differential Pulse Code Modulation, адаптивная дифференциальная импульсно-кодовая модуляция (АДИКМ)
AGP	Accelerated Graphic Port, ускоренный графический порт
ASCII	American Standard Code for Information Interchange, американский стандартный код обмена информацией
ASPI	Advanced SCSI Programming Interface, развитой интерфейс программирования SCSI
AT	Advanced Technology («передовая технология»), класс PC на процессорах 286 и выше
ATA	AT Attachment, интерфейс подключения устройств IDE к компьютеру AT

ATAPI	ATA Package Interface, пакетный интерфейс для ATA (программная спецификация для устройств ATAPI)
BCD	Binary Coded Decimal, двоично-десятичный код
BDA	BIOS Data Area, область данных BIOS в памяти
BIOS	Basic Input/Output System, базовая система ввода-вывода
BIOS INT	BIOS Interrupt, прерывание, обслуживаемое BIOS
BSC	Binary Synchronous Communications, двоичная синхронная передача данных
CAM ATA	Common Access Method ATA, стандарт ANSI, обеспечивающий совместимость IDE-устройств на уровне сигналов и команд
CD	Compact Disk, компакт-диск
CD-ROM	Compact Disk-Read Only Memory, постоянная память на компакт-дисках
CGA	Color Graphic Adapter, цветной графический адаптер
CHS	Cylinder-Head-Sector, цилиндр-головка-сектор, традиционная схема трехмерной адресации данных на диске
CMOS	Complimentary Metal Oxide Semiconductor, комплиментарная структура металл-оксид-полупроводник (КМОП)
CMOS RTC	CMOS Real Time Clock, часы-календарь
COM Port	COMmunication Port, последовательный порт
CPU	Central Processor Unit, центральный процессор
CRC	Cyclic Rendancy Check, контроль с использованием циклического избыточного кода
DAC	Digital-to-Analog Converter, цифроаналоговый преобразователь (ЦАП)

DC 1. Digital Control, цифровое управление (монитором);
2. Direct Current, постоянный ток

DCE Data Communications Equipment, аппаратура передачи данных, например модем (АПД или АКД)

DDC Display Data Channel, интерфейс обмена данными с монитором (для PnP-мониторов)

DIN connector Deutsch Industrie Norm connector, малогабаритный круглый многоконтактный разъем

DIP Dual In-line Package, корпус (микросхемы) с двухрядным расположением штырьковых выводов

DMA Direct Memory Access, прямой доступ к памяти

DPMS Display Power-Management System (Signal), система (сигнал) управления энергопотреблением монитора

DSP Data Signal Processor, процессор обработки сигналов

DTE Data Terminal Equipment, оконечная аппаратура — COM-порт, принтер, плоттер (ООД)

ECC Error Checking and Correcting (Memory), обнаружение и исправление ошибок (в памяти)

ECP Extended Capability (Communication) Port, расширенный LPT-порт для подключения принтеров и сканеров, входит в IEEE 1284

ECU EISA Configuration Utility, утилита конфигурирования устройств шины EISA

EEPROM Electrical Erasable Programmable Read-Only Memory, электрически перезаписываемая постоянная память

EGA Enhanced Graphics Adapter, расширенный графический адаптер

E-IDE Enhanced IDE, расширенный интерфейс IDE

EISA Extended Industry Standard Architecture, расширенная шина ISA

EPP Enhanced Parallel Port, расширенный LPT-порт для подключения внешней памяти и коммуникационных адаптеров, входит в IEEE 1284

EPROM Erasable Programmable Read-Only Memory, стираемая (ультрафиолетовым облучением) программируемая память только для чтения

ESCD Extended Static Configuration Data, расширенные данные о системной конфигурации (в энергонезависимой памяти или на диске); используются для конфигурирования устройств Plug and Play

EVC Enhanced Video Connector, расширенный интерфейс подключения мониторов, включающий дополнительные шины

FCC Federal Communications Commission, федеральная комиссия США по коммуникациям. Сертификат FCC Class B означает малую степень электромагнитного излучения в области радиочастот и пригодность для домашнего использования устройства

FIFO First-In, First-Out, «первым пришел — первым ушел» (метод обслуживания на основе очереди)

FM Frequency Modulation, частотная модуляция

FM Music Synthesizer звуковой синтезатор с частотной модуляцией

GND GrouND, земля, общий провод питания

H High, старший (байт); высокий (уровень сигнала)

IBM PC/AT см. AT

IBM PC/XT см. XT

IC Integrated Circuit, интегральная схема, чип (ИС)

ICU ISA Configuration Utility, утилита конфигурирования унаследованных (Legacy) устройств шины ISA для PnP

ID Identifier, идентификатор (устройства SCSI)

IDC Insulation-Displacement Connector, разъем, смещающий изоляцию

IDE
1. Integrated Drive (Disk) Electronics, устройство (диск) со встроенным контроллером;
2. интерфейс этих устройств (официальное название – ATA);
3. Intelligent Drive Equipment, интеллектуальное оборудование дискового накопителя

IEC International Electrotechnical Committee, Международный комитет по электротехнике (МЭК)

IEEE Institute of Electrical and Electronic Engineers, Институт электротехники и электроники, устанавливающий многие стандарты

IEEE 1284 спецификация режимов LPT-порта (SPP, ECP, EPP и др.)

INT Interrupt, прерывание (вектор прерывания)

IO Input/Output, ввод-вывод

IR-CON Infra-Red Connector, разъем инфракрасной связи

IrDA Infrared Data Association, ассоциация производителей аппаратуры для инфракрасной связи

ISA Industry Standard Architecture, промышленная стандартная архитектура, тип системной шины

ISO International Standardization Organization, Международная организация по стандартизации (МОС)

L Low, младший (байт); низкий (уровень сигнала)

LBA Logical Block Addressing, линейная адресация данных на диске через логический адрес блока

LPT 1. Line Printer, построчный принтер;

2. обозначение параллельного порта для подключения принтера

LSB Least Significant Bit, младший бит

LSI Large Scale Integration, микросхемы с высокой степенью интеграции (БИС, 100–5000 компонентов)

LUN Logical Unit Number, логический номер устройства SCSI

MCA Micro Channel Architecture, микроканальная архитектура (разработана IBM для PS/2)

MDA Monochrome Display Adapter, монохромный видеоадаптер

MIDI Musical Instrument Digital Interface, цифровой интерфейс музыкальных инструментов

MPU-401 UART приемопередатчик MIDI

MSB Most Significant Bit, старший бит

MSI Medium Scale Integration, микросхемы с малой степенью интеграции (ИС из 10–100 компонентов)

NTSC National Television System Committee, американский стандарт цветного телевещания

NV RAM Non-Volatile RAM, энергонезависимая память

OEM Original Equipment Manufacturer, производитель оборудования (в отличие от конечного пользователя)

OS операционная система (ОС)

PAL Phase Alternating Line, построчное чередование фазы, международный стандарт цветного телевещания

PC Personal Computer, персональный компьютер, если не сказано обратное, подразумевается совместимость с IBM PC (ПК)

PC Card стандарт на шину и размеры модулей расширения блокнотных PC (ранее назывался PCMCIA)

PCI Peripherial Component Interconnect bus, шина взаимодействия периферийных компонентов

PCMCIA Personal Computer Memory Card International Association, стандарт на адаптеры блокнотных PC; новое название — PC Card

PGA 1. Professional Graphic Adapter, профессиональный графический адаптер

2. Pin Grid Array, керамический корпус ИС с матрицей штырьковых выводов

PIO Programming Input/Output, программный ввод-вывод

PnP Plug and Play, P&P, «подсоединяй и работай»; стандарт автоматической настройки конфигурации подключаемых устройств

POST 1. Power On Self Test, тест начального включения;
2. Procedure Of Self-Testing, процедура самотестирования

PPM Pulse Position Modulation, позиционно-импульсная модуляция

PROM Programmable Read-Only Memory, однократно программируемая постоянная память

PWM Pulse Width Modulation, широтно-импульсная модуляция (ШИМ)

PXI PCI eXtensions for Instrumentation, расширение шины Compact PCI для инструментальных систем

RAID Redundant Array of Inexpensive Drives (Disks), массив дисковых накопителей с избыточностью (тип дисковой памяти с резервированием и/или дублированием данных)

RAM Random Access Memory, память с произвольным доступом (ОЗУ)

RAMDAC RAM Digital-to-Analog Converter, микросхема видеоадаптера, содержащая ЦАП для выходных сигналов RGB и таблицу палитры (color look-up table) в виде матрицы RAM

RFM Radio Frequency Modulator, радиочастотный модулятор (РЧМ)

RGB Red-Green-Blue, красный-зеленый-синий, базовые цвета монитора

RLE Run Length Encoding, метод сжатия данных (применяется в ECP)

ROM Read Only Memory, постоянная память (ПЗУ)

SCSI Small Computer System Interface, интерфейс малых компьютерных систем

SCSI ID SCSI Identifier, идентификатор устройства SCSI

SDLC	Synchronous Data Link Control, синхронное управление передачей данных
SECAM	Sequence Couleur A Memoire, французский стандарт цветного телевещания, принятый также в странах бывшего СССР
SMART	Self-Monitoring, Analysis and Reporting Technology, технология самонаблюдения, анализа и сообщения состояния, применяемая в современных накопителях на жестких дисках
SPP	Standard Parallel Port, стандартный однонаправленный LPT-порт
SRAM	Static Random Access Memory, статическая память
TAP	Test Access Port, средства доступа для диагностического оборудования
TTL	Transistor-Transistor Logic, транзисторно-транзисторная логика (ТТЛ)
UART	Universal Asynchronous Receiver-Transmitter, универсальный асинхронный приемопередатчик (УАПП)
USART	Universal Synchronous/Asynchronous Receiver-Transmitter, универсальный синхронно-асинхронный приемопередатчик (УСАПП)
USB	Universal Serial Bus, универсальная последовательная шина
VESA	Video Electronics Standards Association, Ассоциация по стандартизации в области видеоэлектроники
VGA	Video Graphics Array, видеографический массив (тип графического адаптера)
VLB	VESA Local Bus, локальная шина VESA
WB	Write Back, обратная запись (алгоритм кэширования)

WT 1. Write Through, сквозная запись (алгоритм кэширования);

2. Wave Table, волновая таблица (содержит оцифрованные фрагменты звуковых сигналов)

WT Synthesizer Wave Table Synthesizer, синтезатор с табличным синтезом

x86 семейство процессоров, совместимых с 8086/8088: 8086, 286, 386, 486, Pentium, Pentium Pro, Pentium II и т. д.

XGA eXtended Graphic Adapter, высокопроизводительный видеоадаптер

XT eXtended Technology («расширенная технология»), класс PC на процессорах 8086

АКД оконечная аппаратура канала данных, например модем (DCE)

АПД аппаратура передачи данных; то же, что и АКД (DCE)

АЦП аналого-цифровой преобразователь (ADC)

БИС большая (по степени интеграции) ИС, содержащая 100–5000 компонентов

ИРПР параллельный интерфейс, отличающийся от Centronics (IFSP)

ИРПР-М параллельный интерфейс, аналогичный Centronics

ИС интегральная схема (чип)

ИУ инициирующее устройство (инициатор)

КЗ короткое замыкание

ЛУ логическое устройство

ОЗУ оперативная память (RAM)

ООД оконечное оборудование данных — источник и/или приемник информации (DTE)

ОС операционная система (OS)

ПЗУ постоянная память (ROM)

ПО программное обеспечение

ПУ периферийное устройство

ТТЛ транзисторно-транзисторная логика (TTL)

ТТЛШ транзисторно-транзисторная логика с диодами Шоттки (TTLS)

УАПП универсальный асинхронный приемопередатчик (UART)

УСАПП универсальный синхронно-асинхронный приемопередатчик (USART)

ЦАП цифроаналоговый преобразователь (DAC)

ЦУ целевое устройство

ЭЛТ электронно-лучевая трубка (CRT)

Алфавитный указатель

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

X

А

В

Г

Д

З

И

К

Л

М

Н

О

П

Р

С

Т

У

Ф

Х

Ч

Ш

Э